Versatile Multidimensional Pattern Analysis for Automated Facial Modeling and Architecture Parsing

Tang, Rui

Award date:
2017

Awarding institution:
University of Bath

Versatile Multidimensional Pattern Analysis for Automated Facial Modeling and Architecture Parsing

submitted by

Rui Tang

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Sciences

September 2017

ABSTRACT

This thesis presents a methodology that demonstrates how 2D image processing techniques can be applied to provide solutions for 3D models. Moreover, for the research sets the aim of evaluating the feasibility and effectiveness of this methodology by its implementation in two specified areas, namely, 3D facial mesh alignment and objects recognition in CAD floor plans, respectively. Regarding the former, an image processing method called optical flow is applied in order to help the registration of 3D meshes. Additionally, the novel algorithm is evaluated by comparing its performance with that of the-state-of-art techniques. In relation to the second application, the aspect of component recognition in CAD floor plans, image processing based automatic system for analysis and labelling floor plans drawn in Computer-aided Design (CAD) form is covered. Experiments comparing the proposed system against other mature systems are evaluated, along with the outcomes of a comprehensive user survey on the novel system being presented.

ACKNOWLEDGEMENTS

First of all, I would like to extend my sincere gratitude to Dr.Darren Cosker, my supervisor, for his instructive advice and helpful suggestions regarding this thesis. Without his patient assistance and friendly encouragement, it would have been impossible for me to complete the work in such a short period of time without reducing its academic value. I have benefited greatly from his critical thinking, wealth of knowledge, scholarly expertise, and friendly encouragement since I started with the research programme. During my research, he inspired me and motivated me to reach my own goals as well.

Secondly, warm tributes must paid to Dr.Kewei Duan, Dr.Wenbin Li, Dr.Chris Li and Dr. Gang Ren, whose profound knowledge and genuine interest my research has greatly encouraged in pursuing my research endeavour.

Thirdly, I want to express my thanks to all the people that I met at University of Bath and University of York, Prof. Peter Hall, Prof. Will Smith and so on, who gave me a warm response when I needed help. Without them, it would have been hard for me to have succeeded in completing my research journey.

Moreover, I want to thank my friends for their very patient and time-consuming assistance in data collection and analyses. Needless to say, all their assistance has greatly facilitated the investigation and hence, also enhanced the quality of this research. Any faults remaining in the are my responsibility.

I should finally like to express my gratitude to my beloved parents, who have always been there to help me out of difficulties and have supported me without a word of complaint.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Real world is a highly dynamic world and full of multidimensional data. Such multidimensional data provides a rich source of information for us to understand the world and present thoughts efficiently. For instance, humans use three dimensional cues to extract facial expressions, distinguish underlying emotions and to understand the context. It is also acknowledged that multidimensional information plays a fundamental role in visual behaviour learning.

The field of computer vision aims to extract useful descriptions from images for understanding the real world. The most significant progress, i.e. image process, has been widely applied in static image analysis and understanding. Image processing normally refers to digital image processing, which refers to using mathematical operations for signal processing, for which the input is an image, a series of images, or a video, such as a photograph or video frame. Whilst the output of image processing may be either an image or a set of characteristics or parameters related to the image [68]. It has demonstrated a strong ability to discover the hidden knowledge from lower dimensional data. However, handling multidimensional information is still challenging to current image process systems, in terms of the large amount nature and difficulty in representation. One missing capability is synchronising lower dimension representation with the higher dimensional information of a scene.

In this context, this thesis is aimed at improving automatic analysis of multidimensional information by bridging the two dimensional representation with higher dimensional information in image processing, as well as evaluating the feasibility and effectiveness of this methodology by its implementation in certain

areas of 4D facial data analysis and automatic structure digitalisation. Regarding the former, I improve 4D facial animation by linking the motion of a set of 3D facial meshes to 2D facial images, using optical flow estimation, an image processing method. I further investigate structural scene recognition, with such scenes having a different hierarchy structure to facial data. They are often complex in terms of spatial dimensions and hard to analyse. More specifically, I propose a novel image processing based system for recognising ambiguous assets like walls, doors and windows in a digital floor plan.

To summarise, this thesis involves studying the feasibility and effectiveness of a methodology that examines whether 2D image processing approaches in a two-dimensional aspect are able to represent efficiently and improve multi-dimensional problems pertaining to such as 3D animation and structural assets recognition. This is followed by a conclusion and discussion on further potential work in the field.

1.1 Contributions

This thesis involves applying interdisciplinary research. On the one hand, it requires comprehensive knowledge of image processing, whilst on the other, knowledge and acute perception regarding other research areas, such as mesh analysis and CAD analysis are required. Given the primary problem domain defined previously, the motivation of this research is to prove that the proposed my methodology that, in some certain research areas, algorithm fused with image processing method brings significant improvements compared to those single subject applications. In detail, I will discuss the motivation in two aspect: 3D facial mesh alignment (Published in [139]) and objects recognition in CAD floor plan (Chinese Patent-201610556348.8), respectively.

1.1.1 Three-dimensional Facial Mesh Alignment

Facial analysis has received a significant amount of attention during the last two decades. With the development of acquisition techniques for capturing 3D data from the real world, the problem of estimating the dense correspondence of the vertexes between face meshes has become very topical. The traditional method of aligning meshes in a 3D aspect is Iterative Close Point (ICP), as proposed by Besl [21] in 1992, which simply iteratively finds the closest point between meshes.

In 2010, Myronenko [112] proposed a probabilistic method, the-state-of-art, to align two meshes by maximising the likelihood of an updated source mesh. Both of the methods ignored the other main feature of 3D meshes, namely, colour information.

In Chapter 3, a detailed introduction is given of a novel hybrid registration method aimed at improving the efficiency and accuracy of analysis of the dense correspondence between meshes of D3DFACS [40]. D3DFACS is a 4D facial expression dataset that provides more than 200 verified dense 3D face action unit sequences captured in a relevantly high frame rate (60 fps). However, adjacent facial meshes do not correspond with each other, which leads to extra difficulties for animation. To solve this problem, the raw 3D meshes are projected into a 2D image plane, which is followed by application to a global hybrid non-rigid registration. The achieved correspondent information of the 2D image is further used to align the raw meshes in 3D. Furthermore, taking into account a critical issue regarding the storage of a 4D face mesh dataset, a novel compression algorithm is also developed for 3D facial meshes. At the end of this chapter, a novel facial retargeting framework is introduced.

1.1.2 Component Extraction in CAD Architectural Drawings

Architectural drawings, usually in forms of floor plans, are essential for describing, designing and guiding a construction project. These drawings are often created along an orthographic top-down projection (floor plan), as well as consists of various architectural elements e.g. walls presented as straight lines. Systems for the analysis of such floor plans may focus on 3D model extrusion, with rules given by experts, rather than automatic image processing.

For example, in the early work of Clifford So and his colleagues [135], they put their effort into a semi-automatic method that extracts vectors from a CAD drawing. Their method involves expert rules for wall extrusion, object mapping as well as ceiling and floor contraction. Similar to Clifford So's approach, Lu et al. [103] proposed a system for constructing models from vectorised floor plans. However, their method is still limited to general rule collection on vectors.

In Chapter 4, I develop an image processing based method for automatically recognising such high dimensional floor plans, which provides strong floor plan regularisation and component extraction with dimensional constraints. In this

context, there are several critical problems in the process of recognising and extracting objects from floor plans. Firstly, in practice, the input data are often noisy and wild, for example, a real user may draw architectural elements, like windows, doors and walls, in a non-standard way. In addition, drawings may be presented in various units, drawing styles and dimensions. Inspired by previous studies, I propose an unsupervised system with higher dimensional constraints in order to analyse and label real floor plans. Within this method, a novel fusion strategy involving 2D image processing is introduced by interleaving expert rules and 2D element manipulation. Compared to the-state-of-art algorithm introduced by Lu [103], the proposed system shows advantages in both accuracy and performance, according to scientific evaluation and a user study.

1.2 The Structure of the Thesis

The remainder of the thesis is structured as follows:

1.2.1 Chapter One - Introduction

In chapter one, I state the central argument and the motivation. This chapter also gives a brief introduction to the current literature and the contribution the proposed methodology makes to the subject areas.

Chapter Two - Literature Review

A comprehensive review of relevant research in the different research areas is provided. That is, since the methodology spans two specific research areas, the related works both in the alignment of 3D meshes and the recognition systems for extracting components in CAD floor plans are covered.

Chapter Three - A 3D Mesh Registration Method

In chapter three, inspired by the proposed methodology, I present a novel method fused with an optical flow algorithm for constructing a densely corresponded dynamic 3D facial expression model, in which a hybrid non-rigid registration technique is introduced. Also, an implementation to recover 3D shapes from 2D images is demonstrated. Furthermore, a novel 3D data compression algorithm, which represents 3D face meshes in linear formation, is shown. After that, a realtime facial retargeting framework in demonstrated. At the end of this chapter, further potential work is discussed.

Chapter Four - A CAD Floor Plan Extraction System

In chapter four, I present an image based automatic system for the analysis and labelling of floor plans drawn in the Computer-aided Design (CAD) form. In order to detect and recognise components such as walls, doors and windows in the CAD floor plans, the proposed system involves novel application of a fusion strategy. Firstly, a general rule based filter parsing method is adopted to extract effective information from the original floor plan. Subsequently, an image-processing based recovery method is applied to correct the information extracted by the first step. In order to evaluate this novel fully automatic floor plan analysing system, it is evaluated on a public website, which, on average, archives more than ten thousand effective uses per day.

Chapter Five - Conclusion

Chapter five summarises the contributions of the thesis, and explains how this methodology has improved current systems/methods provided in the focal research areas. There is also discussion on the potential wider application of the proposed approach and suggestions are put forward for possible future research avenues.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews the literature on 3D facial mesh alignment and the developments regarding the recognition of CAD floor plans. Firstly, a review of the literature on the algorithms and applications of aligning 3D facial meshes is presented. Subsequently, a number of component detection techniques for CAD floor plans are reviewed.

2.1 The Acquisition and Registration of 3D Facial Meshes

Machine learning on human behaviour has being a popular topic since the 1990s. A main reason is that its applications have been widely spread into various fields, such as psychology, medicine, entertainment and security [94, 128]. Due to its extensive use, automatic human behaviour understanding is now playing a crucial role in next-generation computing systems [114]. Automatic facial behaviour analysis, including facial expressions of emotion and facial action unit (AU) [55] recognition, has become one of the most popular areas in recent years [110, 70].

The earliest systematic understanding of facial expression is recorded in the expression of the emotions in man and animals, which was published in 1872 by Charles Darwin [44]. Darwin sought to trace the animal origins of human characteristics, such as the pursing of the lips in concentration and the tightening of the muscles around the eyes in anger as well as efforts of memory. In 1971, six primary emotions, that is, happy, sad, surprise, fear, disgust and anger, were

postulated by Ekman and Friesen [54]. Subsequently, they put it forward the Facial Action Coding System (FACS) in 1997 [55]. In the past, before Suwa's work [105], facial expression analysis was commonly regarded as a research subject for psychologists. He introduced a preliminary proposal on automatic facial expression analysis from image sequences, which brought facial expression into the domain of computer analysis. Among some pioneering efforts by the computer science community, Mase [86] proposed one of the earliest automatic facial expression recognition systems in his work of the early 1990s. He applied optical flow in order to detect and estimate facial muscle actions, which can be recognised as facial expression. Following his work, scientists started their study of facial expression by analysing 2D images and many 2D face databases were built. However, until recently, the majority of the available data sets of faces are limited by size, containing only deliberately posed affective displays recorded under highly controlled conditions [128]. Recently, there has been an increasing amount of research on recognising complex emotions (multiple action units) rather than single basic emotion. However, due to the disadvantage of 2D datasets, most of those systems are still highly sensitive to illumination, occlusions and faces. Single-view 2D analysis is incapable of fully describing the information expressed by faces. Alternatively, 3D data with geometric and texture information provides a solution to this problem. For example, in the case of AU recognition, subtle differences between AU18 (Lip Pucker) and AU10 + AU17 + AU24 (Upper Lip and Chin Raising And Lip Presser) are hard to distinguish in a 2D frontal view. However, this can be easily identified from a 3D capture. Along with the GPU and storage technique progress, the acquisition of 3D facial structure and motion has become a feasible task.

In this section, the focus is on the 3D (static 3D) and 4D (dynamic 3D) datasets for facial expression recognition and analysis. Firstly, the acquisition methods for 3D/4D datasets are introduced. The advantages and disadvantages of several different acquisition methods are discussed. Some existing methods on the registration of 3D and 4D face alignments are demonstrated, in particular, tracking and finding dense correspondences, which are the most challenging parts of build such datasets. Furthermore, different methods for registration in both 2D and 3D are explained. Finally, reviews of existing dynamic 3D facial databases are presented.

2.1.1 Acquisition of 3D and 4D faces

The acquisition technique for capturing 3D data is one of the most important for building a 3D facial dataset, as the equipment and method used can affect the quality of the imposition on the subject. A variety of devices and techniques can be employed for acquisition, and in this section the three most commonly adopted methods, namely, single image reconstruction, structured light technologies, and stereo reconstruction algorithms that include photometric stereo and multi-view stereo, are discussed.

Single Image Reconstruction

Reconstruction of a 3D face from a single 2D facial image of low resolution has been a popular topic in computer vision during the last decade [6, 85]. Compared with 3D recording, this can be achieved in unconstrained environments with a conventional 2D camera or mobile phone, while the subjects could be completely unaware, a technique that has high potential in facial behaviour research. However, as these 3D faces are generated from 2D by machine learning, the meshes results do not have high accuracy, which means that they are incapable of studying subtle expressions and facial muscle movements.

3D Morphable Model (3DMM) [24, 25, 125, 30, 151, 115, 56, 133, 6] is one of the most prominent techniques used for reconstructing the 3D facial mesh with texture information from single or multiple 2D facial images captured in an unconstrained environment. Fig.2-1 shows an example of such reconstruction, with this methodology being first presented in [24]. Recently, in [116], a new 3D morphable model with more subjects and higher resolution has been made publicly available. To build such a model, numbers of scans of human faces are obtained from 3D laser scans. Then, those 3D faces are registered by using their pixel intensities and 3D geometric information. This model-based methodology represents a novel face in an image by modelling coefficients of the geometric and texture parameters, which also provide a reconstruction of the 3D shape. In order to generate a 3D face mesh from a 2D image, a probabilistic method is adopted to estimate both parameters. It has been demonstrated that the 3D Morphable Model can be very efficient for extracting 3D facial surface and texture from a single image. [128] However, even in the recent research achievements, these probabilistic estimations still require good initialisation of important parameters

(including lighting direction and pose) and are sensitive to occlusion. Moreover, this method by using a static 3D model is incapable of reconstructing faces with subtle facial details, such as wrinkles or furrows. In general, 3DMM reconstruction from 2D images through the 3D is a convenient way to achieve low accuracy 3D surfaces.

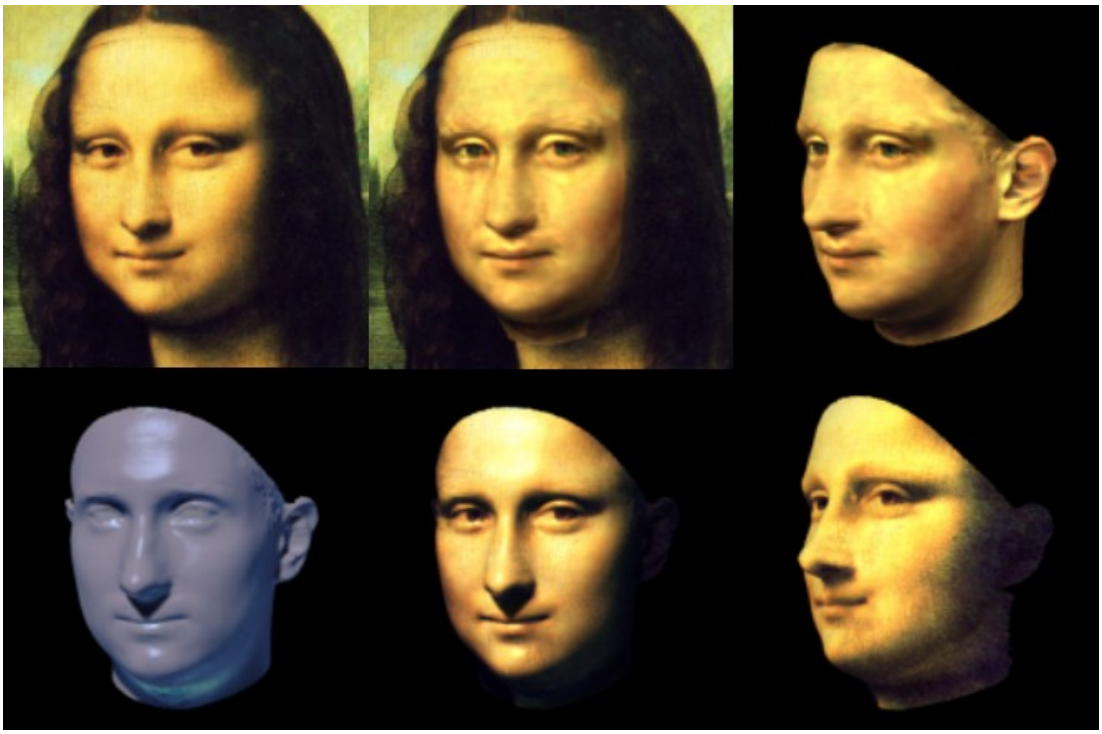


Figure 2-1: *Reconstructed 3D face of Mona Lisa [24] using single image reconstruction*

Structured Light

Structured light method [80, 22, 35, 165] is another common method used for acquisition of 3D face meshes. In order to extract shape information, this technique projects one or more encoded light patterns onto faces and then measures the deformation on the surface of objects, as shown in Fig.2-2. The key step of this technique is switching rapidly between coloured patterns and white light. Thus, in addition to getting shape information, the texture information from human faces is also obtained. However, due to the difference of reflexivity on face skin, hair or beard, the result may contain holes with missing points or small artefacts. Also, this method is restricted in terms of the amount of movement of face in the scene, which has to be in the area simultaneously covered by the structured pattern and visible camera. Despite these disadvantages, by using a single pat-

tern, the structured light 3D face acquisition system always has the capability of obtaining a 3D surface in real-time [63, 79]. Also, compared to a laser scan, the cost of this methodology is much lower, for only a projector and a high-speed camera are needed. Moreover, another main advantage is that, in most cases, the visible projected pattern does not distract the users, because a human-being is unable to perceive it, because of the high-speed, but the projected channel will still appear as a full colour image.

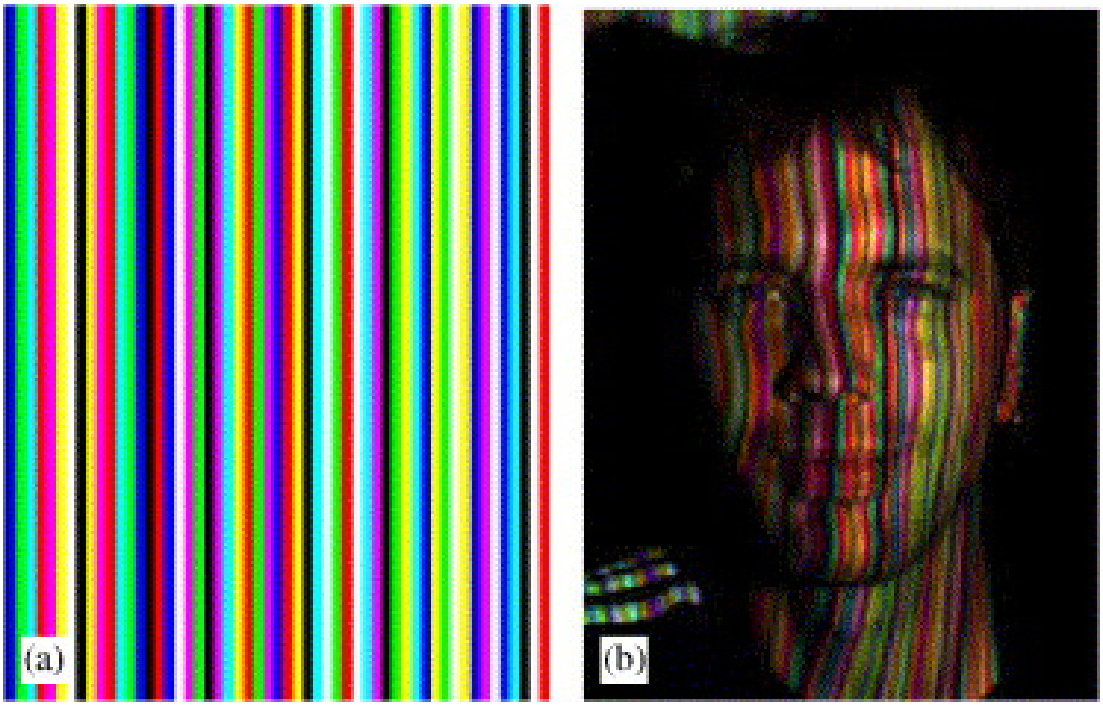


Figure 2-2: *Example of structured light [144]: a. The color pattern used. b. The pattern projected onto a subject.*

Photometric Stereo

Another popular technique for obtaining a 3D structure is photometric stereo, which was first proposed in [157]. Photometric stereo, also known as shape from shading, estimates the orientation field (normals) of a 3D surface of objects from a set of images of objects under different illuminations. A set of results of this method produced by four standard illuminations is shown in Fig 2-3 [31]. Due to this feature, photometric stereo is sensitive to the presence of projected shadows, highlights and non-uniform lighting [128]. Moreover, this methodology obtains 3D normals of objects in the first place, rather than 3D geometric information. Hence, further analysis that computes geometric information from these normals

Figure 2-3: *Production from photometric stereo [31]*

has to be processed. This procedure increases the computation time and introduces additional errors [59, 3, 64, 2, 134]. Furthermore, as lighting has to be changed during the capturing procedure, normally, subjects have to close their eyes when capturing, which is certainly unnatural. Despite these disadvantages, like the structured light method, photometric stereo is a very economic implementation for 3D surface acquisition.

Multi-view Stereo

Multi-view stereo acquisition is the other type of stereo method for 3D facial reconstruction, which is also widely used [131]. In this method, multiple calibrated cameras are placed at specific viewpoints from the subject and thus, same numbers of images of the scene will be captured simultaneously. Based on those simultaneous images, the corresponding points can be found. Then, subject to some constraints, these points can be used for reconstruction. Moreover, in such a system, only the cameras need to be calibrated, for there is no calibration requirement in relation to subjects. In addition, it does not need flashing lights, as all the cameras can record the same scene simultaneously, with constant light sources. Due to this fact, this kind of system does not bring an unnatural experience to users like photometric stereo or the structure light system, and allows more natural behaviour from subjects. Hence, this technique has been successfully employed to develop commercial 4D facial expression systems, such as the DI4D [47] and the 3DMD dynamic 3D stereo system [1] as well as some research programs like [40], [87], [17] and [106]. In such systems, 3D capturing is performed with high-quality equipment in order to provide high accuracy

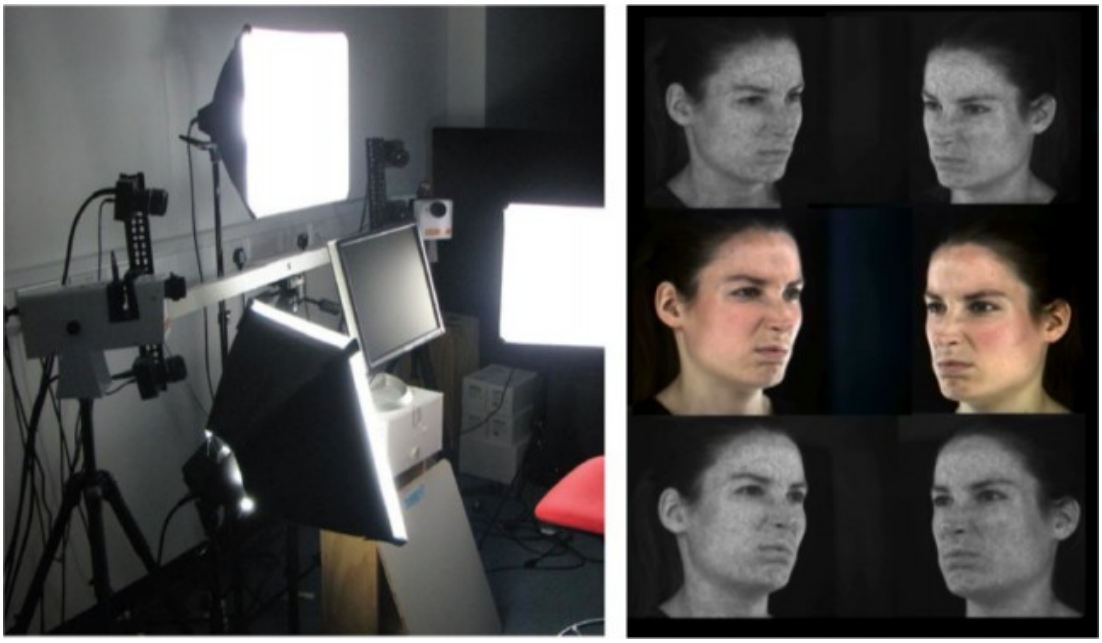


Figure 2-4: *An example of multiple-view stereo acquisition system [40]*

facial geometric results, as shown in Fig 2-4. However, this system still has a high restriction on the head movement of subjects. Compared with photometric stereo and structure light system, it is much more expensive to implement, as several high quality cameras have to be employed. Furthermore, usually, off-line computation, which produces the range maps of subjects through passive stereo photometry method [75], has to be processed. Hence, it is argued here that it is infeasible for real-time systems to use this technique.

2.1.2 3D Mesh Alignment

Accurate alignment and tracking methods on 3D meshes are very important for facial expression systems. Feature based corresponded datasets rely on areas, and always fail to find the movement of specific points on the face, which means a detailed study cannot be performed with such datasets. However, dense correspondence between face meshes allows the scientist to track the full motion of face mesh between the subject or frames. In order to tackle these problems, approaches in both rigid registration and non-rigid registration have been proposed. Iterative close point (ICP) [21] is the most widely used method for rigid alignment, because those meshes are similar in shape without large transformation, while another probabilistic method Coherent Point Drift (CPD), introduced by [112], also addresses the accurate result on rigid alignment. For facial expression study, non-rigid registration method has to be employed. As 3D objects can be projected into 2D, this problem is discussed in both the 2D and 3D aspects in following part.

2D Non-Rigid Registration

From the perspective of computer vision, optical flow estimation is a widely used technique [16, 28] for describing image motion happening in a sequence of images. Based on gray-level images, this methodology computes displacements for feature points or even every single pixel within the image and estimates where these features or pixel could be in the next image frame. However, this technique is limited regarding the brightness between images [125], which means massive additional error will be introduced, if the brightness changes between frames. Moreover, as this method is designed for aligning images with a tiny time step [65], it cannot handle a large degree of rigid transform or an immediate large change in a non-rigid one.

Another widely used 2D registration method is the Active Appearance Model (AAM) introduced by [38]. This method builds a model which learns from a training set that is usually a sequence of annotated images. During the training, some landmarks have to be placed manually at the boundary and some obvious face features, which should be found in all the images of the training sequence. Accordingly, AAM cannot provide dense correspondences across images and thus, (TPS) [27] is employed to do collaboration work [40].

3D Non-Rigid Registration

Iterative close point (ICP) [21] is a simple algorithm that has been widely used for 3D rigid alignment issues [162, 67, 111]. To align two meshes, it iteratively revises the transformation to minimise the distance between two point sets. This algorithm performs well in rigid registration, but additional methods, such as [8] are required. It introduce a stiffness value which controls the rigidity of the transformation that can be applied at each iteration. While processing, this value decreases in order to allow for progressively non-rigid transformations to be applied. However, as a feature of ICP, it requires the initial position of the raw point sets be adequately close. Also, it is vulnerable to noisy data as it will try to fit to all points.

Coherent Point Drift (CPD) [112] is a probabilistic method suitable for both rigid and non-rigid registration. It converts the alignment of two point sets to a probability density estimation problem. However, in the case of an extremely dense and large distributed dataset, it is observed that non-rigid CPD does not perform ideally. Nevertheless, it is able to produce accurate rigid alignment results.

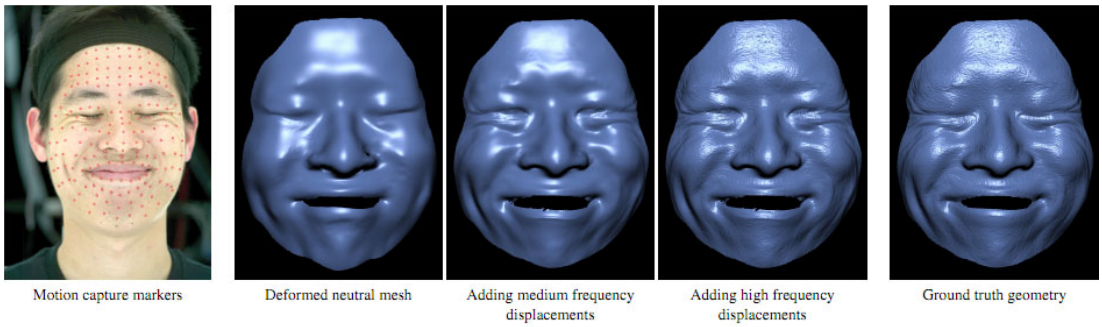


Figure 2-5: *A synthesis result of Ma's Polynomial Displacement Maps method [106]*

Subsequent to these traditional 3D non-rigid registration algorithms, Klaudiny and Hilton [87] introduced a patch based registration method, where a minimum spanning tree is applied. In their system, a non-sequential traversal of the sequence is calculated using the minimum spanning tree based on the dissimilarity of feature locations. Then, a patch based frame to frame non-rigid registration is applied to calculate dense correspondence between pairs of meshes. Additionally, in 2008, another state-of-the-art dense corresponded 3D faces capture and constructing system was proposed by [106] (Fig 2-5). This training based system provides returning high-resolution corresponded mesh (semi-synthesis) results in real-time. However, in order to track the correspondence, a number of facial markers have to be involved in both the training and constructing stages. Then, in 2011, Wilson [155] introduced a method which computes surface correspondences between 3D facial scans in different expressions. By introducing the concept of Active Visage, their technique is able accurately to correspond high-resolution face meshes, which may have wide differences in expressions without requiring intermediate pose sequences. However, this algorithm is semi-automatic, which means user interactions are needed. Given such a drawback, the algorithm is not suitable for application when constructing a dynamic 3D dataset. Moreover, Beeler proposed an optical flow based markerless 3D registration algorithm for 3D face sequences [17]. In this work, the concept of anchor frames was introduced, which are those frames that contain similar facial expression to a manually chosen reference expression. The algorithm, firstly, computes pixel matches directly from the reference frame to all anchor frames. Then, by using the anchor frames to partition the sequence into clips and independently matching clips, the tracking result shows much more robust in bound drifts, occlusion and motion blur (Fig 2-6).

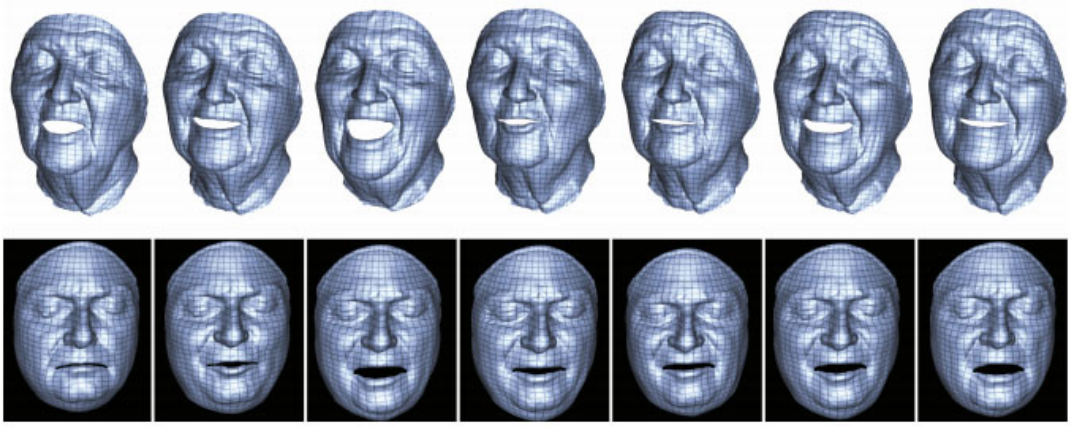


Figure 2-6: *A registration result of Beeler's anchor frame method [17]*

2.1.3 Facial Action Coding System

Facial expression is widely used to evaluate emotional impairment in neuropsychiatric disorders. Ekman and Friesen's Facial Action Coding System (FACS) [54] encodes movements of individual facial muscles from distinct momentary changes in facial appearance. Unlike facial expression ratings based on categorisation of expressions into prototypical emotions (happiness, sadness, anger, fear, disgust, etc.), FACS can encode ambiguous and subtle expressions and therefore is potentially more suitable for analysing small differences in facial affect.

Ekman and Friesen proposed the Facial Action Coding System (FACS), which is based on facial muscle change and can characterise facial actions that constitute an expression irrespective of emotion. FACS encodes the movement of specific facial muscles called action units (AUs), which reflect distinct momentary changes in facial appearance. In FACS, a human rater can encode facial actions without necessarily inferring the emotional state of a subject and hence, it is possible to encode ambiguous and subtle facial expressions that are not categorisable into one of the universal emotions. The sensitivity of FACS to subtle expression differences was demonstrated in studies showing its capability to distinguish genuine and fake smiles [46], the characteristics of painful expressions [118, 41, 117, 124], and depression [121]. FACS has also been used to study how prototypical emotions are expressed as unique combinations of facial muscles in healthy people [54, 69, 89] and to examine evoked and posed facial expressions in schizophrenia patients versus controls [88], which revealed substantial differences in the configuration and frequency of the AUs in five universal emotions.

As above mentioned, Ekman and Friesen developed the Facial Action Coding System (FACS) for describing facial expressions by action units (AUs). Of the 44 FACS AUs that they defined, 30 AUs are anatomically related to the contractions of specific facial muscles: 12 are for the upper face, and 18 for the lower. AUs can occur either singly or in combination. When AUs occur in combination they may be additive, in which the combination does not change the appearance of the constituent AUs, or nonadditive, in which their appearance does change. Whilst the number of atomic AUs is relatively small, more than 7,000 different combinations have been observed [74]. In sum, FACS provides the descriptive power necessary to describe the details of facial expression.

Commonly occurring AUs and some of the additive and nonadditive AU combinations are shown in Figure 2-7 and 2-8. As an example of a nonadditive effect, AU 4 appears differently depending on whether it occurs alone or in combination with AU 1 (as in AU 1 + 4). When AU 4 occurs alone, the brows are drawn together and lowered. In AU 1 + 4, the brows are drawn together, but are raised due to the action of AU 1. AU 1 + 2 is another example of nonadditive combinations. When AU 2 occurs alone, it not only raises the outer brow, but also often pulls up the inner brow, which results in a very similar appearance to AU 1 + 2. These effects of the nonadditive AU combinations increase the difficulties of AU recognition.

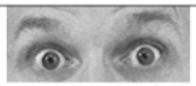
NEUTRAL	AU 1	AU 2	AU 4	AU 5
				
Eyes, brow, and cheek are relaxed.	Inner portion of the brows is raised.	Outer portion of the brows is raised.	Brows lowered and drawn together	Upper eyelids are raised.
AU 6	AU 7	AU 1+2	AU 1+4	AU 4+5
				
Cheeks are raised.	Lower eyelids are raised.	Inner and outer portions of the brows are raised.	Medial portion of the brows is raised and pulled together.	Brows lowered and drawn together and upper eyelids are raised.
AU 1+2+4	AU 1+2+5	AU 1+6	AU 6+7	AU 1+2+5+6+7
				
Brows are pulled together and upward.	Brows and upper eyelids are raised.	Inner portion of brows and cheeks are raised.	Lower eyelids cheeks are raised.	Brows, eyelids, and cheeks are raised.

Figure 2-7: *Upper Face Action Units and Some Combinations [142]*

NEUTRAL	AU 9	AU 10	AU 12	AU 20
				
Lips relaxed and closed.	The infraorbital triangle and center of the upper lip are pulled upwards. Nasal root wrinkling is present.	The infraorbital triangle is pushed upwards. Upper lip is raised. Causes angular bend in shape of upper lip. Nasal root wrinkle is absent.	Lip corners are pulled obliquely.	The lips and the lower portion of the nasolabial furrow are pulled pulled back laterally. The mouth is elongated.
AU15	AU 17	AU 25	AU 26	AU 27
				
The corners of the lips are pulled down.	The chin boss is pushed upwards.	Lips are relaxed and parted.	Lips are relaxed and parted; mandible is lowered.	Mouth stretched open and the mandible pulled downwards.
AU 23+24	AU 9+17	AU9+25	AU9+17+23+24	AU10+17
				
Lips tightened, narrowed, and pressed together.				
AU 10+25	AU 10+15+17	AU 12+25	AU12+26	AU 15+17
				
AU 17+23+24	AU 20+25			
				

Figure 2-8: *Lower Face Action Units and Some Combinations [142]*

2.1.4 Dimension Reduction

In machine learning and statistics, dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration [126], via obtaining a set of principal variables. It can be divided into feature selection and feature extraction [119].

Principal Component Analysis

The main linear technique for dimensionality reduction, principal component analysis, performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximised. In practice, the covariance (and sometimes the correlation) matrix of the data is constructed and the eigen vectors on this matrix are computed. The eigen vectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. Moreover, the first few eigen vectors can often be interpreted in terms of the large-scale physical behaviour of the system. The original space (with dimension of the number of points) has been reduced (with data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors.

Isomap

Isomap [140] is a combination of the Floyd Warshall algorithm [58] with classic Multidimensional Scaling. Classic Multidimensional Scaling (MDS) takes a matrix of pair-wise distances between all points, and computes a position for each point. Isomap assumes that the pair-wise distances are only known between neighbouring points, and uses the Floyd Warshall algorithm to compute the pair-wise distances between all other points. This effectively estimates the full matrix of pair-wise geodesic distances between all of the points. Isomap then uses classic MDS to compute the reduced-dimensional positions of all the points.

Manifold Alignment

Manifold alignment [82] takes advantage of the assumption that disparate data sets produced by similar generating processes will share a similar underlying manifold representation. By learning projections from each original space to the shared manifold, correspondences are recovered and knowledge from one domain

can be transferred to another. Most manifold alignment techniques consider only two data sets, but the concept extends arbitrarily to many initial data sets [150].

2.1.5 Analysis by Synthesis

Analysis by synthesis is a scientific method by which one attempts to understand (or analyse) a phenomenon by reconstructing (or synthesising)it, often in a computer simulation [60]. In the research area of computer vision and graphics, referring to the exposition in Li's Book [95], it is the process that aims to analyse a signal or image by reproducing it using a model. The objective of analysis by synthesis is to find the value of the model parameters that synthesise the closest image possible in the span of the model. It is then an optimisation problem that requires the setting of a cost function (e.g. sum of squares) and of a model with a small number of parameters. The model must be able to generate typical variations (such as pose, illumination, identity and expression for face images), to enable the analysis of a signal or image that includes expected variations.

Analysis by synthesis is widely applied in face matching and facial recognition. For example, in the research of [24, 128, 52, 23, 76], scientists have applied the analysis-by-synthesis method to solve face matching problems. Briefly speaking, they compare between an enrolment image A and an image A_0 which is synthesised from an input probe image in such a way that the image properties of the latter image resemble those of the former.

2.1.6 Dynamic 3D Databases

In the last two decades, a number of 3D face databases have been developed in order to perform facial analysis, such as face modeling and recognition. In this subsection, drawing on the survey work provided by [128] (Fig.2-9), the three publicly most accepted 4D (dynamic 3D) models are reviewed, namely: Change's database [35], BU-4DFE [20] and D3DFACS [40].

Change's database is the first 3D facial expression dataset that includes six subjects that express the six basic facial expressions. The data were obtained through the structured light method, as described in Section 2.1.1. However, there is no landmark or dense correspondence between frames and the database is not publicly available.

BU-4DFE [20] was created in order to assess the individuality of facial motion for person verification. 3D faces in the dataset was obtained using the 3DMD

Name	S/D	Size	Content	Landmarks	Annotation	P
Chang et al. [21]	D	6 adults	6 basic expressions	N/A	N/A	N
BU-3DFE [75]	S	100 adults	6 basic expressions at 4 intensity levels	83 facial points	N/A	Y
BU-4DFE [51]	D	101 adults	6 basic expressions	83 facial points for every frame	N/A	Y
Bosphorus [76]	S	105 adults inc. 27 actors	24 AUs, neutral, 6 basic exps, occlusions	24 facial points	25 AUs	Y
ICT-3DRFE [77]	S	23 adults	15 exps: 6 basic, 2 neutral, 2 eyebrow, 1 scrunched face, 4 eye gaze	N/A	AUS with intensity levels	Y
Tsalakanidou et al. [61]	S	52 adults	11 AUs and 6 basic expressions	N/A	N/A	N
Benedikt et al. [52]	S	94 adults	Smiles and word utterance	N/A	N/A	N
D3DFACS [53]	D	10 adults inc. 4 FACS experts	Up to 38 AUs per subject	N/A	AU peaks	Y
Blanz Vetter [10,18]	S	200 adults	Neutral faces	N/A	N/A	Y
ND-2006 [78,79]	S	888 adults	Neutral and 5 exps: H, D, Sa, Su, random	N/A	N/A	Y
CASIA [80,81]	S	123 adults	Neutral and 5 exps: smile, laugh, A, Su, eyes closed	N/A	N/A	Y
Gavdb [82]	S	61 adults	3 exps: open/closed smiling and random	N/A	N/A	Y
York 3D [83,84]	S	350 adults	Neutral and 4 exps: H, A, eyes closed, eyebrows raised	N/A	N/A	Y
Texas [85,86]	S	105 adults	Neutral and smiling, or talking with open/closed eyes	25 facial points	N/A	Y

Figure 2-9: *3D face databases containing expression data [128]*

Face Dynamic System [1]. It contains 101 adult subjects, while each of them has several basic expressions. Nevertheless, this database is feature base corresponded rather than dense corresponded.

D3DFACS [40] is the only FACS coded dynamic 3D facial AU system, which contains 10 subjects obtained by using 3DMD. For each subject, up to 38 different expressions was posed with a single AU or multiple ones, which were coded by four FACS experts. Moreover, all 519 facial expression sequences in the database were captured at 60 frames/sec, comprising approximately 90 frames for each. It is the first database that will allow for research into dynamic 3D AU recognition and analysis [128]. However, this database is neither feature nor dense corresponded.

2.2 Analysis on Architectural CAD Floor Plans

2.2.1 Architectural Floor Plan

Architectural drawings, usually in forms of floor plans, are vitally necessary in designing, describing and performing a construction project. Since the architectural elements in each building level are represented by using standard symbols, floor plans normally create an orthographic top-down projection.

Floor plans consist of various levels of detailed architectural elements. Taking construction structure drawings(CSDs), the most complicated floor plan, as an example, these portrays internal steel bars, the concrete structure for columns, beams and WA walls, and pipe and ductwork layouts. Tong Lu [103] and his research team introduced a system that constructs a detailed building model from computer-drawn CSDs.

Whilst the floor plans that are widely used in the architecture engineering and construction life cycle are able to cover a building's complete layout, it is impossible to deny the fact that they lack detailed construction information.

Another key drawback comes from the various graphic symbols deployed in these plans. A drawing's motivation, not being constrained to a specific standard, determines what components will be shown and how. Despite the fact that less-detailed floor plans can be regarded as legitimate input by many systems, the various symbols still create challenges when analyzing and interpreting their image.

2.2.2 Analysing Floor Plan CAD files

Systems analysing CAD-based floor plans focus more on 3D model extrusion rather than image processing and pattern recognition.

Berkeley researchers Rick Lewis and Carlo Sequin [92], from the University of California, introduced a system that creates 3D polygonal building models semi-automatically by grouping architectural symbols into specified layers in standard DXF files. In order to overcome geometric flaws, this system corrects disjoint and overlapping edges when recognising the algorithm's task. In the process, it collects the topology of spaces and portals to generate proper polygon orientation. After each floor has been modelled, the system piles the floor and thereby, creates the complete model. This system was a great improvement since it simplifies the recognition process, which benefits designers in various applications, such as

smoke propagation simulation.

At Hong Kong University of Science and Technology(HKUST), Clifford So and his colleagues [135] put their efforts into viewing the model conversion problem in the VR context. Clifford's team targeted three major tasks, including: wall extrusion, object mapping, and ceiling and floor contraction after observing conventional manual model reconstruction. The processing time was reduced greatly by incorporating automated approaches to each task in the next step, such as automatic wall polygon extrusion, generating and placing customised templates of random orientation and size, and advancing front triangulation. The system Clifford proposed has a significant disadvantage in that the input file must contain fully established semantic information and no errors, which means it requires manual effort to be put in. For example, wall lines must be marked up by users, architectural objects must be specified and objects must be assigned to individual transformation matrices.

Researchers at the Massachusetts Institute of Technology (MIT) started to automate construction of a realistic campus model in the Building Model Generation (BMG) project [26] (http://city.csail.mit.edu/bmg). Compared with the Berkeley system, the pipeline is similar, while an extra process is attached in order to position and orient building models automatically using a map for guidance.

Lu's research team [103], in the Nanjing University of China, proposed a system for constructing models from computer-drawn CSDs and vectorised floor plans. Compared to a computer drawing, a vector image is much more difficult in symbol recognition, because it contains geometric primitives without labels for type indicating. This system shares things in common with the HKUST project in relation to differentiating the walls from other architectural elements. Firstly, it detects parallel line-segment pairs as walls, which are then removed from the drawings. Next, the remaining primitives are recognised by detecting feature matches with predefined patterns that contain symbols, graphical primitives and contexts. In the process of recognition, the system places the patterns in order, according to their priority level and checks them one by one. Corresponding elements are removed from the drawing as soon as they meet all of a pattern's constraints. Whilst it requires high quality input, the system benefits users greatly, because it not only focuses on structural details, for it also is a highly automated process.

2.2.3 Image Parsing and Drawing Analysis

Overviews

As a specific task, floor plan analysis has been addressed over 20 years, with the purpose of extracting the layout information and detecting architectural elements by analysing an input raster floor plan.

Tracing back to previous research, [10] developed a prototype system of understanding a hand-sketched floor plan, which converted the drawing into CAD format automatically. Through scanning, processing, extraction and identification, this system interpreted hand-sketched floor plans into CAD formats by describing architectural elements. This system benefited designers substantially as manual conversion had been replaced completely. After being tested on 150 realistic floor plan drawings, this system proved to be outstanding, because elements identification was finished within 35 seconds at that time. However, with technology developing, processing efficiency needed to be improved as well. Another drawback was that hand-sketched drawings gradually faded out, being replaced by computer-produced drawings.

By sharing a similar purpose, a system that targeted to understand hand-sketched floor plans was proposed in [99]. This system was improved based on [10], with the methods of subgraph isomorphism and Hough transform [51] being adopted. By doing this, the process of matching was enhanced significantly.

Understanding a hand-sketched architectural drawing consists of recognising building elements and structural properties. The recognition process is usually completed by applying the Hough transform (SLHT) based method, where either walls or other building primitives can be identified. Introducing isomorphism in order to recognise the reminder of the graph is applied in the following step.

With the purpose of reconstructing in 3D buildings, a complete system is introduced in [50]. Starting with image processing and feature extraction from drawings, the next step of this system is to convert it into an initial 2D modelling in the form of basic architectural entities. In the next step, a 3D modelling process is proposed to match the reconstructed floors.

The complete system has advantages over others for following reasons. To start with, it consists of a number of automated graphic recognition processes and most notably, it integrates a flexible user interface.

Different to previous research, [107] pays great attention to room detection in architectural floor plans. In this system, the first main step is to extract the

Step	Issues
Noise removal	The leading lines of notations could be easily confused with wall lines. The background might contain a grid or decorative pattern.
Text extraction	Text font, size, and orientation might vary. Text and graphical symbols might share pixels(overlapping or touching)
Vectorization	Most algorithms recover only lines and arcs. Free-form curves continue to be a challenge. Noise greatly affects the result. Vectorization might give bad results at junction points.
Symbol recognition	The symbols might not comply with the standards. There might be a large pool of symbols, and differences between two symbols could be subtle.

Figure 2-10: *Issues in Floor Plan Recognition*

primitives in the drawings, where the lines and arcs constitute the walls and doors, respectively. It also presents the ways that doors are hypothetically detected by extracting arcs. The next step focuses on detection of rooms in buildings.

The Hough Transform has been valuable in the line detection process. Then, the walls in the floor plan are able to be located by deducing other graphical prosperities.

[5] categorised the floor plan analysis system into three parts: information segmentation, structural analysis, and finally, semantic information extraction and alignment. This system was evaluated by drawing on the extant literature and through experiments. The results from experiments on a large corpus of 90 floor plans are positive.

Challenges

Drawing on a 2009 survey [163], the challenges of image parsing and drawing analysis are explained clearly in Figure 2-10 as follows.

In the process of image parsing and drawing analysis, cleaning, vectorisation and graphical-symbol recognition are three major steps. Generally, in most systems, the analysis process starts with cleaning that is aimed at improving recognition quality by removing noise and unnecessary information. In the following step, graphical-symbol recognition, the system is applied to category recognised symbols by identifying information including location, orientation and scale.

Compared to other graphical documents, floor plans have features that can be distinguished from others. Firstly, varied shapes of lines, either curved or straight, represent walls in floor plans. Another difference is the architectural symbols are made up of simple geometric primitives. Typically, in order to deal with this type of input, graphics recognition is usually integrated with vectorisation. The cleaning process consists of noise removal and text extraction, following by vectorisation and symbol reorganisation.

Noise Removal Sampling noise brought by digital scanning is one of the most common types when processing hand-sketched floor plans. However, since floor plans are gradually generated by computer, noise has a broader definition in addition to sampling noise. For example, pixels without directly useful information usually are considered as noise, including annotation leading lines, dimension lines, furniture, and hardware symbols. On rare occasions, the decorative pattern in the background can become confused as well.

Related to Loria system, a morphological filter is applied as a fine line between noise and useful pixels. This method is based on the assumption that the background patterns and dimension leading lines can be picked out from useful lines, because they are different in thickness and style. [113] provides a similar assumption, filtering input so only thick construction lines can be preserved.

Text Extraction An assumption perfect algorithm should be free from text font, size, and orientation, along with having the advantages of efficiency and little manual intervention. Geometric shapes mixed with text put extra burden on separation and extraction. Text research has been developed for several decades, and the results can be categorised into two groups: structural-based algorithms (focus on structural difference) and pixel based.

Graphic Recognition Text is separated from graphics in the previous step. Graphic recognition is the process where pixels are organised and put in order

in geometrical description of the building layout. Usually, architectural drawings consist of two major types of information: structural information and local architectural components.

As shown in Table 2 above, graphic recognition is composed of vectorisation and symbol recognition. Walls are preserved as geometric polylines for the extrusion step, since they define the building's spatial structure. With this consideration, all systems introduce vectorisation and deal with geometric elements instead of performing symbol recognition on pixels directly.

Vectorisation This process aims to transfer image pixels to the geometric primitives, so it is also called raster-to-vector conversion. The most important standards of each algorithm are efficiency, robustness, and accuracy. The work flow of traditional line-drawing vectorisation contains two steps as shown in the following table.

Fixing joint errors is required after each step above. In most cases, vectorisation algorithms are able to find out line segments and circular arcs. However, more complex curves are still a challenge to algorithms.

In the step 1, three groups of algorithms are usually used by systems, including parametric model fitting, contour tracking and skeletonisation [77]. In parametric model fitting, the Hough transform is applied to detect lines, but this has the drawbacks of over-consumption of memory and lack of universality.

Contour tracking is an algorithm that detects the contour of white pixels (instead of black ones) and recognises connected regions as rooms. This method is able to deal with simple floors, but not those with complicated structures, because it is based on the assumption that white spaces are divided by wall lines that are represented as black in the image.

Thinning-based algorithms of skeletonisation are intended to thin/search for a curve bones' medial axis by stripping boundary pixels until a one-pixel-wide skeleton remains [90]. One of its disadvantages is that intersections always confuse the results. Another one, is they take a long time, since each pixel is visited more than once. Typical medial-axis-based algorithms include pixel tracking [48] and run-graph-based algorithms [48]. Medial-axis-based algorithms treat a thick line as a solid shape and its medial axis as a skeleton.

In step 2, point chains are segmented into sets of lines, polylines, and circular arcs by estimating curvature or polygonal approximation and then, finding out the critical points.

Loria's system introduces a skeletonisation technique and polygonal approximation to complete the vectorisation process. The CUHK system tracks the contour of the black pixels rather than the white ones, which is different to contour tracking.

Symbol Recognition This is the most important part in graphical document analysis, with the graphic symbol recogniser (GSR) in this process expected to be efficient and limited to neither context nor affine transformation (Previous research has proven that several methods work well in certain types of CAD drawings and generate positive results.

Generally, there are two types of GSRs that are widely accepted: vector based (oriented toward structure) and pixel based (oriented toward statistics). Vector based GSRs process graphical primitives such as points, line segments, arcs, and circles in vectorised images. This approach checks primitives in groups in order to identify a symbol, which includes region adjacency graph [100], graphical-knowledge-guided reasoning [158], constraint networks [4], and deformable templates [148]. Good vectorisation is expected and it is affine invariant.

The other GSRs are pixel-based, which process raster images without vectorisation being involved and they focus on the statistical features of a symbol's pixel information. The approaches contain plain binary images [130], living projection, and shape contexts [19]. Compared to vector-based approaches, a pixel-based one can generate higher accuracy, although its performance is sensitive to scaling and rotation. Su Yang [160] involved himself in improving the recognition method by merging these two approaches.

In Loria's project [50], a network is applied to identify features of a vectorised image's primitives, and then segments in a vectorised floor plan are distributed throughout the network, with the aim of finding terminal symbols. A similar, but simpler, approach is introduced in CUHK's system, in which a series of geometric constraints is regarded as symbol patterns. Either raster or vector copies of a floor plan are accepted by the systems to improve recognition accuracy.

Other Systems

Currently, the most popular systems for extracting structural objects from CAD floor plans used worldwide are integrated into AutoCAD Revit [123] and Chief Architect [37]. However, according to their manuals, users are able to extract walls, windows and doors only if they landmark the lays manually. Consequently,

it is hard to define them as fully automatic recognising systems.

CHAPTER 3

AN IMAGE PROCESSING BASED REGISTRATION METHOD FOR ALIGNING A DYNAMIC 3D FACIAL DATASET

3.1 Introduction

Facial analysis in 3D has been a salient topic in the computer vision and graphics community. A significant reason is that facial analysis in 3D is more robust to pose and light variance, especially for face identity and facial expression recognition. There are many datasets which can be used for static 3D facial expression analysis. Moreover, dynamic 3D facial expression models have recently become popular due to the robust factors described above. However, in contrast to the multiple 2D facial expression datasets, there are only two publicly available dynamic ones in 3D, namely: BU-4DFE [161] and D3DFACS [40]. The former is mainly used for facial expression recognition, whilst the latter is designed for AU (Action Unit) analysis. Nevertheless, neither of these dataset is in dense correspondence, a crucial feature that is required in order to track the full motion of the face mesh between subjects or frames. One task of my PhD research is accurately tracking the vertexes between face meshes and constructing a dense corresponded dynamic 3D facial expression model. In this chapter, the key contribution in this area from my PhD study is presented and then, some possible directions for further research are proposed.

First, the focus is on the dense correspondence estimation of the vertex between face meshes, in particular, the mesh non-rigid registration technique, based

on mesh pairs. A novel non-rigid registration method is proposed to process and construct a dense corresponded dynamic 3D facial dataset. In experiments, D3DFACS is used, in which facial expression sequences are encoded by FACS experts [40], as raw data for registration and modelling. When aligning face meshes, major challenge in doing so is that human face meshes are typically based on multiple recordings of different facial expressions (Action Units), as opposed to multiple subjects, e.g. neutral expressions of different people [24, 116]. Furthermore, in the raw dataset, the first frames of a face mesh (defined as neutral frame) in each AU sequence,contain a high degree of rigid head movement and they are not ideally neutral, which means there are some vestiges of facial expression on those frames. Since raw data in a dynamic 3D dataset is obtained by motion capture, there are normally over 10k frames. Hence, the other difficulty is that the registration method has to be fully automatic, devoid of manual adjustment. Iterative close point (ICP) [21] and some algorithms based on ICP [8] are simple and traditional 3D mesh rigid and non-rigid registration algorithms, which have been widely used for 3D alignment issues. However, ICP and its derivative require the initial position of the raw point sets to be adequately close. Also, they are vulnerable to noisy data, as they will try to fit to all points. Hence, a novel non-rigid registration method is proposed, in which the action unit registration to the global and local phases is split for processing the dynamic 3D morphable model (in which 3D data are dense corresponded) construction. Additionally, evaluation of the proposed method compared with other methods, such as CPD and ICP, is provided.

In the remaining parts of this chapter, I move to some further study of the proposed dynamic model. At first, I implement an efficient linear method [6], which is capable of recovering a full 3D shape of faces from single 2D images. This linear algorithm shows excellent results in reconstructing 3D shapes from synthetic 2D face images, however, the reconstruction result on real 2D faces is not entirely satisfactory. In addition to the implementation of the 2D to 3D fitting, a novel modelling and compressing algorithm analysing dynamic 3D face expression data is put forward.

In the current literature, the problem is that face expression data in 3D (especially dynamic 3D data) is relatively expensive in terms of both storage and computing. Some dimension reduction methods, such as principal component analysis(PCA) have been utilised to address this problem [24]. However, the length of the eigen vector in PCA constrains not only the accuracy of recon-

struction, but also the size of the training dataset that has to be large enough to produce a precise model. In contrast to PCA, the proposed compressing method is able to reduce the representation of each sample down to one dimension, while carrying more temporal information. Additionally, for the purpose of reducing mesh into one dimension, several optional compressing returns are further involved in the method put forward. In order to evaluate the proposed algorithm, firstly, the performance is quantitatively measured against PCA. In addition, experiments concerning the spatial-temporal aspects have been processed.

In addition to the research on a 3D dynamic model and 3D non-rigid registration, I participated in the evaluation of a project about Optical Flow Estimation Laplacian Mesh Energy [96]. Moreover, I also contributed and designed a building rank system for another collaboration project about building synthesis using part base model recombination, which will be submitted to Siggraph.

The rest of this chapter is organised as follows: Section 2.1 explains the background including the history of facial analysis, acquisition of 3D facial data, registration of 3D meshes and the existing 3D datasets. Section 3.2 analyses current problems concerning my PhD research and some possible solutions are proposed. In Section 3.3, a novel non-rigid registration method, which processes the dynamic 3D morphable model, is put forward. Furthermore, in Section 3.4, a proposed compressing algorithm for a dynamic 3D dataset and its evaluation are presented. Then, Section 3.5 shows a realtime facial retargeting framework in demonstrated. The chapter concludes with a discussion on further work in Section 3.6.

3.2 Problem Statement

As mentioned in Section 2.1, several techniques for dense alignment of 3D meshes have been proposed and studied, e.g. Iterative close point (ICP) [21], Coherent Point Drift (CPD) [112], the Active Appearance Model (AAM) [38], Polynomial Displacement Maps [106], Active Visage [155] and Anchor Frames [17]. There was also a brief review of three existing dynamic 3D facial expression databases (Change's database [35], BU-4DFE [20] and D3DFACS [40]). However, for various reasons, no dense corresponded dynamic 3D facial expression model has been developed.

According to Section 2.1, after projecting 3D meshes onto a 2D plane, optical flow performs well on dense pixel registration, however, it is sensitive to a large rigid transform. Meanwhile, the numerical iteration method, nonrigid ICP, is vulnerable to noisy data as it will try to fit to all points. Moreover, algorithms based on one probability study, such as CPD, only work well with a small point set. So, in the case of an extremely dense and large distributed dataset, the result from CPD is not strictly acceptable. Different from these traditional methods, Polynomial Displacement Maps, Active Visage and Anchor Frames are all capable of producing ideal 3D registration results. Nevertheless, Polynomial Displacement Maps need facial markers in both the training and producing stages and user interactive is required in Active Visage, while Anchor Frames are designed for long sequences that have frames that are similar with the reference one.

Given these facts, in Section 3.2.1, the key difficulties to developing such a model that could perform 3D registration are listed below. Then, in Section 3.2.2, solutions for tackling these difficulties are discussed.

3.2.1 Key Difficulties

The key difficulties when developing a dense corresponded dynamic 3D facial expression model are:

a. How do we obtain high quality dynamic 3D facial expression samples?

b. From raw data, how do we remove variation in pose and head movements?

c. What kind of automatic non-rigid registration method can be used to align large amounts of meshes in dynamic 3D dataset? How should the registration within sequences, between sequences and between individuals be handled?

3.2.2 Technique Solution

In this section, solutions are proposed for addressing the research difficulties raised in Section 3.2.1:

a. Since [40] have made their high-quality dynamic 3D facial expression database public available, these raw data are used directly;

b. Rigid registration, such as CPD can be applied to normalise raw meshes into a standard space;

c. An optical flow based hybrid non-rigid registration method for dense alignment in sequence and between sequence is introduced in the next section, which is capable of addressing this problem. This approach involves registration of meshes in sequence and between sequences as local and global issues, respectively. In order to implement the alignment, local registration is scheduled first and thereafter, global alignment is processed. More details are provided in the next section.

3.3 Global Alignment

In this section, a novel non-rigid registration method to process a dynamic 3D morphable model (in which 3D data are dense corresponded) construction based on Cosker's FACS data [40] is proposed. As previously explained, one difficult aspect of using such a dataset is that human face meshes are typically based on multiple recordings of different facial expressions (Action Units), as opposed to multiple e.g. neutral expressions of different people [24, 116]. Another problem is that, in the raw dataset, the first frames of face mesh (defined as neutral frame) in each AU sequence are not ideally neutral, which means there are some vestiges of facial expression on those frames. Moreover, each expression sequence contains a large degree of rigid head movement. Hence, as mentioned in Section 3.2, the proposed algorithm introduces a novel hybrid registration method to track dense 3D correspondence in the raw dataset. After defining a global reference frame, the method involves applying a raw rigid alignment on all the frames in the dataset. Then, a 3D to 2D projection is adapted to give each 3D mesh a 2D representation in gray-level. Based on the 2D gray images, an optical flow based hybrid non-rigid registration method is employed to estimate the correspondence between a pair of 3D frames. Due to the size of the 3D meshes, once all have been fully corresponded, a statistic approach is introduced for data compression. The approach is subsequently evaluated by comparing it with CPD.

The background to 3D face data modelling and the salient recent research are briefly presented in Section 3.3.1. In Section 3.3.2, the strategy of rigid registration is outlined in detail. Subsequently, in Section 3.3.2, how to project 3D face meshes onto a 2D plane is explained and the framework of the optical flow based hybrid non-rigid registration is present. Additionally, Section 3.3.2 provides evaluation of the proposed method.

3.3.1 Introduction

In recent years, facial analysis using 3D models has been a central topic in computer vision and graphics. The main reason of this is that such models are more robust to pose and light invariance in recognition, allowing for the estimation of 3D facial shape from 2D images [115]. To benefit from these advantages, the 3D morphable model (3DMM) was introduced by Blanz and Vetter [24] in 1999. This statistical model has been widely used to perform various tasks, such as face recognition [25], expression transfer between individuals [116] and reconstruction

of a 3D face from 2D images [133].

Facial expression recognition is also an active research area, with many works being based on recognising facial movement descriptions according to the Facial Action Coding System(FACS) [53]. As previously explained, FACS was primarily introduced by psychologists to describe different configurations of facial actions or Action Units (AUs). It provides 44 individual AUs, which form the basis of 6 prototypical facial expressions: happiness, sadness, fear, surprise, anger and disgust. Despite FACS recognition being widespread in 2D facial analysis, there are only a limited number of 3D facial datasets available, and only one dynamic 3D facial expression database based on FACS [40]. Consequently, in this section, a statistic dynamic 3D FACS data sets, which is comparable to the state of the art in 2D, is proposed. Generally, the contribution of this work can be summarised in four respects, as follows.

Three-Dimensional Representation

Each individual face with one or multiple AUs can generate a set of diversity 2D images, which makes analysis difficult. On the other hand, facial databases in 3D are more robust to pose, illumination and expression during analysis. For this purpose, the dynamic 3D FACS dataset is designed to support presenting face scans with texture in a 3D coordinates system.

Correspondence-Based Representation

Correspondence-based representation is an essential feature for face databases to be powerful in facial analysis. By registering different scans into a same vector space, this allows for generation or description of a face by the linear combination of several faces. In order to build such a model, raw face scans are registered to a common space. Compared to Blanz and Vetter, who archived this by processing an optical flow approach on a image pair for dense correspondence estimation, their accuracy is improved upon here by employing a probability based registration method, Coherent Point Drift (CPD) [112], to cooporate with Optical Flow Estimation Using Laplacian Mesh Energy (LME), which is one of the state-of-the-art optical flow algorithms [96].

Facial Action Unit Representation

Facial expression recognition and speech analysis have become increasingly popular in the last decade. As a result, FACS is widely used in the study of facial expression understanding. A lot of experiments [127, 97] have been carried out to classify these AUs in both 2D and 3D. Given this fact, each face expression sequence in the proposed model was able to be AU coded by FACS experts.

Statistic Representation

As mentioned before, correspondence-based representation has enabled the description of a face in a linear combination. However, its coefficients are not in a normalised space and hence, Principle Component Analysis (PCA) has been carried out to learn the distribution. Additionally, despite using massive 3D data, PCA reduced the dimension of each mesh to a real-time computable entity.

3.3.2 3D Dynamic Morphable Model

Sixty-three AU sequences were extracted from the D3DFACS data set for one participant, with the proposed approach being used to build a 3D dynamic morphable model for this person.

Difficulties of 3D Registration

The registration problem between to meshes or surfaces is one of most basic problems in computer vision and computer graphics. For this problem, two sets of 3D points are given and the task is to align optimally these two sets of points by estimating a best transformation between them [93]. Due to the importance of this issue, it arises as a subtask in many different applications (e.g. object recognition, tracking, range data fusion, graphics, medical image alignment, robotics and structural bioinformatics etc. [149, 78, 83, 156, 66, 109]).

The major challenge of registration on 3D meshes is that the point-wise correspondences between the two point sets is often unknown *a-priori*. Under this situation, the registration problem is also known as the simultaneous pose and correspondence problem (SPC problem) [93].

According to the explanation of [9], the main difficulty in the 3D registration problem is determining the correspondences of points on one surface to points on the other. Local regions on the surface are rarely distinctive enough to determine the correct correspondence, whether because of noise in the scans, or because of symmetries in the object shape. Hence, the set of candidate correspondences to a given point is usually large. Determining the correspondence for all object points results in a combinatorially large search problem. The existing algorithms for deformable surface registration make the problem tractable by assuming significant prior knowledge about the objects being registered. Some rely on the presence of markers on the object [136, 7], while others assume prior knowledge about the object dynamics [98], or about the space of nonrigid deformations [91, 24]. Algorithms that make neither restriction [132, 71] simplify the problem by decorrelating the choice of correspondences for the different points in the scan. However, this approximation is only good in the case when the object deformation is small; maxima as nearby points in one scan are allowed to map to far-away points in the other.

3D Rigid Registration

In order to align each individual AU mesh into a normalised space, a rigid registration on meshes in D3DFACS dataset before non-rigid registration is applied. The raw data in the dataset suffer from a noisy out layer and incoherency of rotation issues. The rigid registration to align two meshes is performed as well as possible. According to Rui's method [139], when compared to ICP, CPD is more robust for data with noise and a high degree of rigid variation. As opposed to applying CPD on each individual 3D mesh, firstly, neutral expression mesh in D3DFACS is chosen as the global reference mesh G. Then, for each AU sequence in D3DFACS, the CPD registration method is applied to estimate the transform matrix between the first meshes of each AU sequence and a global reference mesh. Following this, in order to align the remaining meshes in each AU sequence to the global reference, the calculated transform matrix is applied to the remaining frames in the AU sequence. The process is as follows:

a. A neutral expression mesh in D3DFACS is chosen as the global reference mesh G;

b. The first frame of the ith AU sequence is marked as Fi;

c. CPD is applied between Fi and G to obtain a rigid transformation matrix RTi of the ith AU sequence.

d. By applying RTi on each mesh in the ith AU sequence, meshes in the ith are rigid aligned to the global reference mesh G.

3D Global Non-Rigid Alignment

3D to 2D projection After rigid alignment, all the 3D data are normalised into a standard space. Since the non-rigid registration will be processed in 2D image space, it is necessary to project the vertex from 3D space onto a 2D plane. Hence, a cylindrical projection is applied to these data to obtain cylindrical UV maps for each aligned 3D scan. More specifically, as the meshes in D3DFACS are normalised into a uniformed space, a cylindrical projection template for the purpose of projection is built. As the result of such uniformed cylindrical projection, for the ith sequence with meshes $\chi i = [X_{i1}, X_{i2}, \dots X_{in}]$, where $X = [\mathrm{x}_1^T, \mathrm{x}_2^T, \dots \mathrm{x}_m^T]$, $\mathrm{x}_i = [x_x^i, x_y^i, x_z^i]^T \in \mathbb{R}$, a set of UV texture maps $\mathbb{I} = [I_1, I_2, \dots I_n]$, and a set of UV Geometric coordinates $\mathbb{U} = [\mathrm{U}_1, \mathrm{U}_2, \dots \mathrm{U}_n]$,

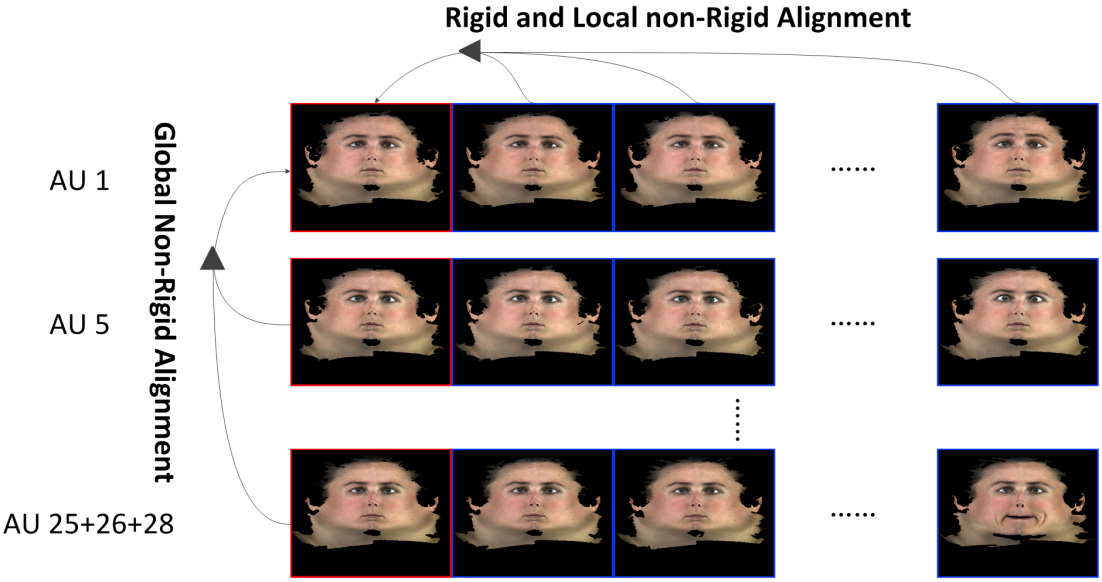


Figure 3-1: *Procedure of Local and Global Registration*

where U= $[u_1^T, u_2^T, \ldots, u_m^T]^T$, and $u_i = [u_i, v_i]$ is generated. Using $\mathbb{U}$ and χ an additional set of *3D images* $I_{3D}(\mathbf{u}) = x_i$ is generated. These map any point in the UV space $u_i = [u_i, v_i]$ to a 3D mesh coordinate $x_i = [x_x^i, x_y^i, x_z^i]^T$, thus allowing for 3D deformation in image space to be handled. More details are described in [40].

Local Optical Flow Based Non-Rigid Registration After rigid alignment of the data set, the next aim is to create vertex correspondences through the set of meshes. Typically, each mesh has a different number of vertices, so the challenge is to deform non-rigidly a reference mesh using pixel tracking. A two step non-rigid alignment process is employed to solve this problem, which first acts locally (intra-sequence) and then globally (inter-sequence).

Firstly, a local alignment is applied to each AU sequence individually. Given the UV texture maps from sequence I_i, the optical flow fields $f_1, f_2, f_3, \ldots f_{n-1}$ between the pairs $(I_1, I_2), (I_2, I_3), (I_3, I_4), \ldots (I_{n-1}, I_n)$ are estimated. These fields are applied to warp both the UV texture maps and 3D images, such that they are all aligned back to the neutral expression of the AU sequence (typically the first frame). The corresponding UV texture maps and 3D images are denoted as $\mathbb{I}_i^{\dot{U}V}$ and $\mathbb{I}_i^{\dot{3}D}$, respectively.

Global Hybrid Non-Rigid Registration After local non-rigid alignment for each individual sequence, a global non-rigid alignment is then employed to align all UV textures and 3D images, for each AU sequence, to the global reference

(global neutral expression image). In this procedure, first, the optical flow fields are computed between the global reference image and the texture reference images of each sequence, respectively. As the difference between the neutral expressions for AU sequences can be large, this can result in a noisy flow field and hence, lead to warping of artefacts. To address this, optical flow is combined with Thin Plate Splines (TPS) [27] in this stage to achieve a more consistent warp field. The global neutral reference assigned a set of landmarks, and the flow field positions of these are used to obtain corresponding landmarks in the neutral references of each AU sequence. Then, TPS is used to warp $\mathbb{I}_i^{\dot{U}V}$ and $\mathbb{I}_i^{\dot{3}D}$ to the global neutral reference face, thus obtaining $\mathbb{I}_i^{\mathring{U}V}$ and $\mathbb{I}_i^{\mathring{3}D}$.

Statistical Model Once the faces are fully corresponded through all the sequences, they can be parameterised as triangular meshes with vertices and the shared topology. Following the assumption of the independence of texture and geometric information in [24], the UV Texture Maps $\dot{\mathbb{I}}_i$ and the UV Geometric Maps $\dot{\mathbb{U}}_i$ are represented in a linear combination using PCA, then:

$$T = \bar{T} + \alpha * P_T, \;\; S = \bar{S} + \beta * P_S \tag{3.1}$$

where, $\bar{T}$ is the mean UV Texture Map and $\bar{S}$ is the mean mesh. P_T and P_S are the eigenvectors of $\dot{\mathbb{I}}_i$ and $\dot{\mathbb{U}}_i$, whilst α and β are vectors of the weights. Then, any new face meshes can be generated by varying α and β, which control shape and texture.

Additionally, by rewriting equation (1), all the meshes can be represented into a lower dimensional space:

$$\alpha = P_T(T' - \bar{T}), \;\; \beta = P_S(S' - \bar{S}) \tag{3.2}$$

where, T' and S' denote a UV Texture Map and UV Geometric Map of any mesh. This lower-dim representation provides a more efficient way for further 3D facial analysis and it also makes a potential use for real-time study.

Evaluation

In this section, the performance of the proposed hybrid non-rigid registration is evaluated and the results are compared with Coherent Point Drift (CPD) [112]. The comparative results are also demonstrated visually.

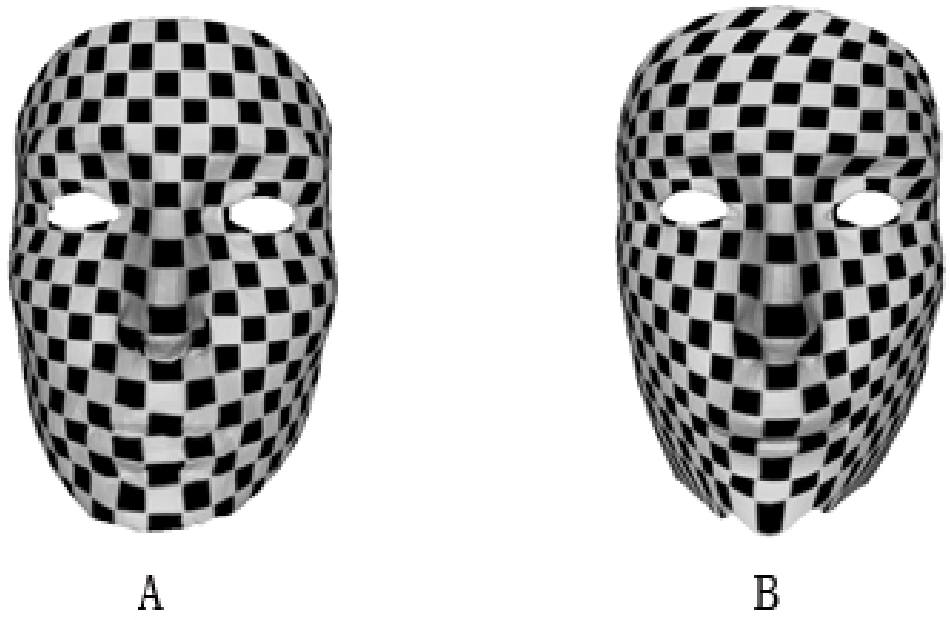


Figure 3-2: *Visual Comparison between LME and CPD. A. Alignment result of LME. B. Alignment result of CPD*

Comparison against CPD First, an evaluation of the proposed hybrid non-rigid registration against Coherent Point Drift (CPD) was performed. More specifically, during the step of local registration, CPD and LME, were applied respectively in order to find the correspondence between mesh pairs. Figure 3-2 shows the visual comparison between LME optic flow and CPD (Action Unit 22, Index 255). According to the visual result, these methods perform equally on the stable area. However, for the edges and area with large variation, there is distinct distortion in the results for CPD.

It is a common concern that there is no ground truth to compare while aligning meshes from raw data. Hence, another new strategy was designed, which compares vertex movement on a specific region (forehead, cheek and nose), as shown in Figure 3-3. In this test, sequences of AU 1 (Inner Brow Raiser), AU 4 (Brow Lowerer), AU 12 (Lip Corner Puller) and AU 22 (Lip Funneler) were sampled. In order to evaluate the proposed registration method, the geometric changes in the key regions are observed. In Figure 3-3, b and c show the average shape of four action unit sequences. The green, blue and red region on the

surfaces represent cheek, nose and forehead, respectively.

Additionally, in sub figures a and c, the average distance of each region from the neutral frame is illustrated. By comparing sub figures a and c, this reveals that the vertex movement in the registration results of LME is well corresponded with the AUs, while that for the CPD results is less so. This is because in the case of extremely dense and large distributed point pairs the parameters of GMM based CPD needs to be precisely adjusted, whilst the LME non-rigid registration is robust for the local registration stage.

Synthesising Expression In this subsection, the results of synthetic expression by using the proposed PCA model (see Fig 3-4) are provided. A combined expression of AU4 (Brow Lowerer) and AU12 (Lip Corner Puller) was synthesised. Firstly, the peak frames in AU4 (Input 1) and AU12 (Input 2) are chosen. Then, by averaging their linear representation in the PCA model, a novel combined expression (result) was reconstructed, where there is no distinct blur, except in the corner of the left lip (marked out by a red ring).

Conclusion and Further Work

In this section, the proposed non-rigid registration method has been implemented to process the dynamic 3D morphable model construction based on the Dynamic 3D FACS Dataset (D3DFACS) [40]. Firstly, the registration problem is divided into two steps: rigid registration and non-rigid registration. Additionally, in non-rigid registration, alignment is split into local registration and global registration. Whilst the local step processes frames in facial expression sequence, the global step processes frames between sequences. In the evaluation comparison has been made between the proposed method and that of the state-of-the-art method, namely, Coherent Point Drift (CPD). Moreover, the result of synthesising expression by using the model has been presented. For future work, in order to improve the alignment result, the aim is to apply the minimum spanning tree method [87] to the current work. Additionally, the interest lies in finding dense correspondence across subjects, thus creating a linear realistic facial expressions synthesising system (details in Section A).

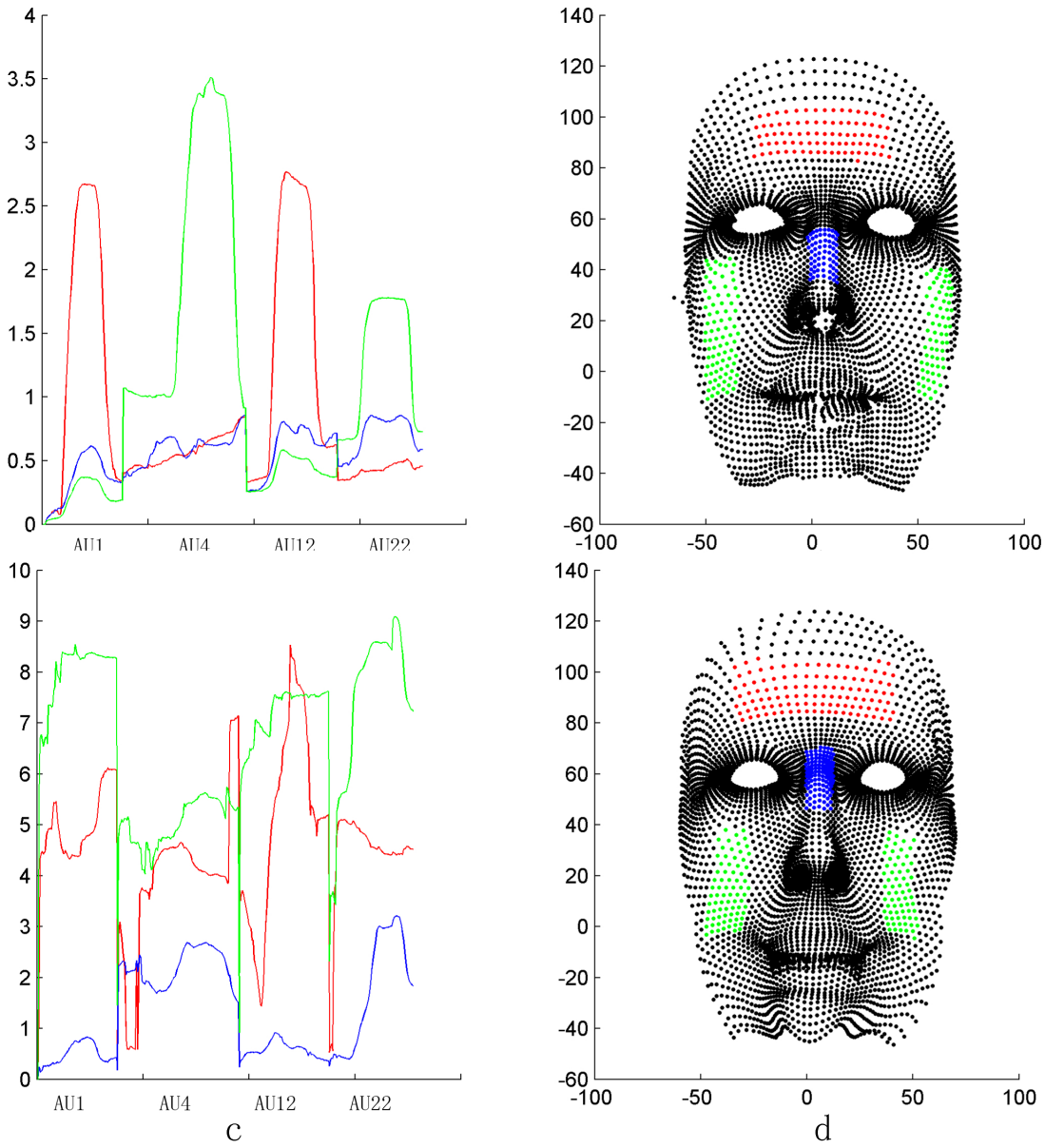


Figure 3-3: *Vertex movement in different regions. LME result : a. distance of the vertex from the reference frame in AU 1, 4, 12 and 22. b. mean shape of LME registration products. CPD result : a. distance of the vertex from the reference frame in AU 1, 4, 12 and 22. b. mean shape of the CPD registration products.*

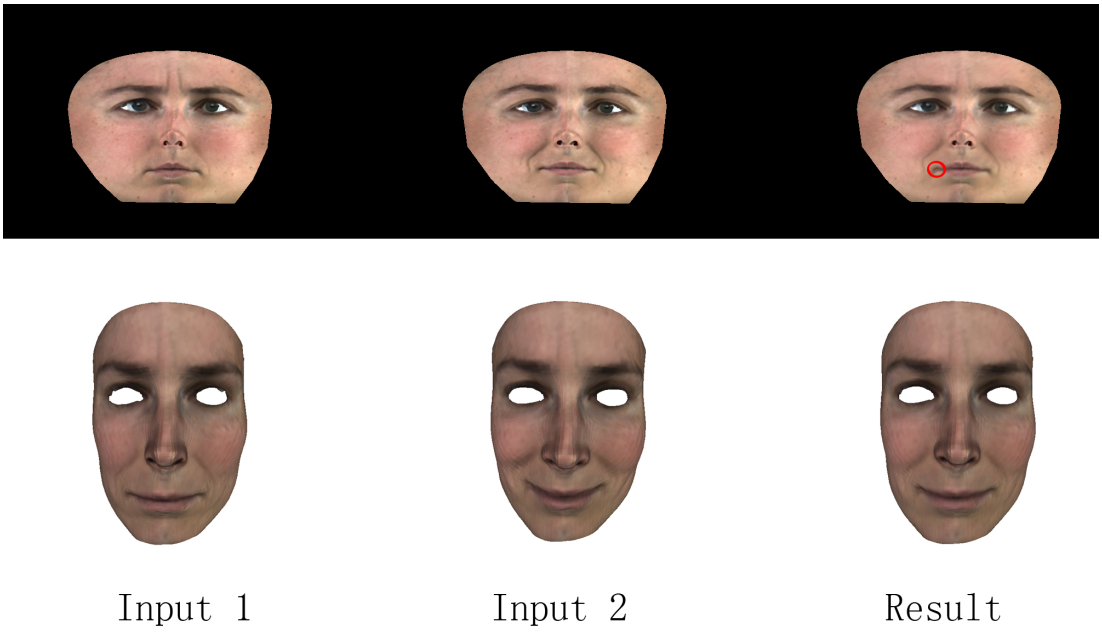


Figure 3-4: *Synthesising Expression. Input1 and Input2 represent the peak frames of AU4 and AU12, respectively. Row1 shows UV texture maps, while Row2 demonstrates the corresponded 3D model.*

3.4 Curve Compression

3.4.1 Introduction

In this section, a novel modelling and compression algorithm for analysing dynamic 3D face expression data is proposed. In the extant literature, the problem is that face expression data in 3D is relatively expensive in terms of both storage and computing. As a common solution, the correspondence between meshes is tracked and then some dimension reduction method, such as principal component analysis(PCA), is utilised to address this problem [24]. However, the length of the eigen vector in PCA limits the accuracy of reconstruction and the size of the training dataset has to be large enough to produce a precise model. In contrast to PCA, under the proposed method the representation of each sample can be reduced down to one dimension. Moreover, the product of PCA shows low temporal correlation, while that of the proposed algorithm is highly spatially-temporally related. In the following subsections, the compression algorithm will be explained, with experiments and evaluations also being presented. At the end of this section, some potential future works are suggested.

In recent decades, principle component analysis has been widely used for dimensionality reduction in a variety of areas, especially in facial analysis. One factor is that it is easy for implementation, the other is that PCA produces only small errors after compressing and decompressing. The operation mechanism of PCA is numerically based and the compression result has a low spatial and temporal relation. However, facial analysis, such as capturing dynamic 3D facial expression and blend shape, is video based, and the data are highly temporally related.

Hence, it is necessary to compress such data through both spatial and temporal terms. It is contended that the proposed novel compression method has better performance than PCA in terms of the compressing and decompressing errors. Moreover, unlike PCA, this method keeps temporal information after compression. As for the alignment method in Section 3.3, Dynamic 3D FACS Dataset (D3DFACS) [40] is used in the experiments.

The proposed algorithm will be explained in two stages:

A. A linear compression method is introduced;

B. Supportive experimental results are presented.

3.4.2 Compression Algorithm

By applying the alignment of all 3D meshes with a reference mesh (Section 3.3), the vertices in all the meshes are corresponded. Subsequently, the corresponded meshes can be compressed by the proposed linear method with the support of the previous alignment step. As aforementioned, for the purpose of compressing meshes down to one dimension, this method can also compress the meshes into alternative sizes. In this subsection, first, the common step of the compression algorithm is presented. Then, the branch steps of Vertex Curve Compression, Clustered Curve Compression and Mesh Curve Compression, respectively, are shown.

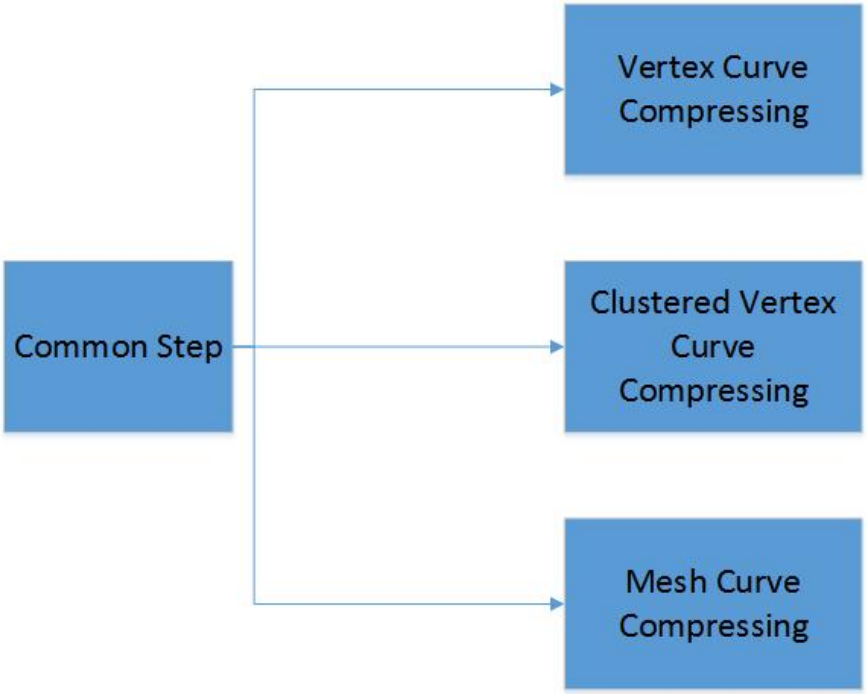


Figure 3-5: *Algorithm Overview.*

Common Step

The first step's objective is to obtain the eigen difference map for each AU sequence. When giving this sequence (e.g. AU12), its neutral frame is named N_i and its peak frame P_i, whilst i is the index of the AU sequence, in this case, $i = 12$. Based on equation 3.3, the difference in the maps regarding N_i and P_i is named the Expression Difference Map E_i.

$$E_i = N_i - P_i \tag{3.3}$$

Then, for any mesh M_{ij} in the AU sequence, there is a difference map D_{ij}, where j denotes the jth frame in AUi.

$$D_{ij} = M_{ij} - N_i \tag{3.4}$$

In E_i, the key information of each AU sequence is stored. In addition, by using D_{ij}, the geometric changes of each mesh are tracked, which are the bases of the branch steps.

Branch Step

Vertex Curve Compression As each corresponded vertex is an individual part of a mesh, then it is able to track the movement of a vertex based on D_{ij} (Figure 3-6). Sub-figure.c is the visualisation change in the AU12 sequence, and sub-figure.a shows the movement of the red dot in the xyz axis. It can be seen that the change of any vertex in a mesh can be represented by three curves and these can be compressed into one (sub-figure.b) by using Equation 3.3.

$$W_{ijk} = E'_{ik} \times D_{ijk} \tag{3.5}$$

E_{ik} is the eigen value of the kth vertex in E_i, D_{ijk} corresponds to its value in D_{ij} and W_{ijk} denotes the weight of the kth vertex in frame j of AUi. Because corresponded vertices are individual parts of a mesh, W_{ij} can be obtained, which stores the weight of all the vertices in the jth frame of AUi.

Afterwards, a normalisation step is applied. Based on Equation 4, the weights of vertices in the neutral frames and peak frame are always 0 and 1, respectively.

$$W_{ijk} = \frac{W_{ijk}}{E'_{ik} \times E_{ik}} \tag{3.6}$$

Finally, based on E_i and D_{ijk}, each vertex in a mesh can be represented by W_{ijk}, according to Equation 5, where M_{ijk} is a subset of M_{ij}.

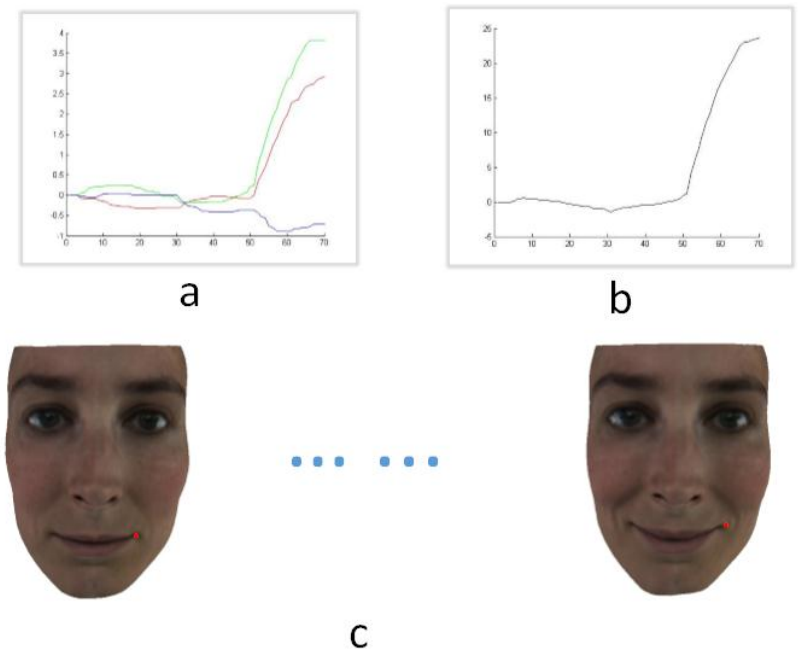


Figure 3-6: *a. The movement of the red dot in the xyz axis in AU12. b. Compressed curve of the original xyz curve. c. Visualisation change in the AU12 sequence*

$$M_{ijk} = N_{ik} + W_{ijk} \times E_{ik} \tag{3.7}$$

In this case, the geometric information of a mesh can be compressed to one third of its original size and an AU sequence, which has n vertices in each mesh, can be represented by a given neutral frame N_i, an Expression Difference Map E_i and n curves W_i.

Clustered Curve Compression

In Section 3.4.2, the movement of a face can be represented by a number of curves, where each curve is a movement trajectory of the vertex over time. However, the movements of the vertices are not independent of each other. To overcome this issue, the curves are grouped using EM (Expectation-Maximization) based clustering toolbox [61]. By applying this, curves that are correlated with each other are members of the same group. In the experiment, K is set as 3 to cluster n curves into three groups. Figure 3a shows the curve clustering result (when K=3) for AU12. Figure 3b is the representation of the clustering result in the

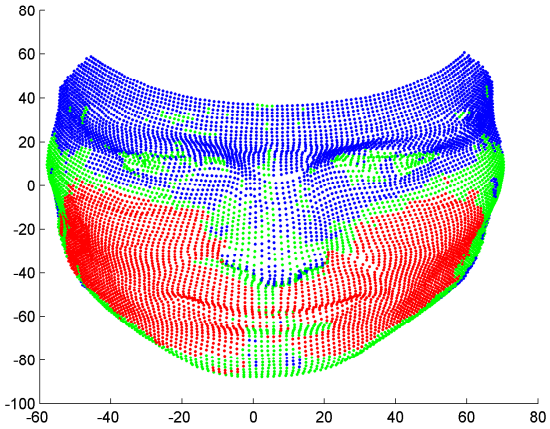


Figure 3-7: *Clustering result on a 3D face: red, green and blue represent the different regions classified*

visualisation aspect. Similar to Section 3.4.2, all the curves in each group are compressed into one curve by following equations:

$$\mathbb{W}_{ijk} = \mathbb{E}'_{ik} \times \mathbb{D}_{ijk} \tag{3.8}$$

$$\mathbb{W}_{ijk} = \frac{\mathbb{W}_{ijk}}{\mathbb{E}'_{ik} \times \mathbb{E}_{ik}} \tag{3.9}$$

$$\mathbb{M}_{ijk} = \mathbb{N}_{ik} + \mathbb{W}_{ijk} \times \mathbb{E}_{ik} \tag{3.10}$$

$\mathbb{E}_{ik}$ and $\mathbb{N}_{ik}$ denote Group$k$'s subsets of Expression Difference Map E_i and neutral frame N_i, respectively. $\mathbb{W}_{ijk}$ is the weight value of the vertices in Group k. In the above equations, Equation 6 calculates the weight curves for the vertices in group k, Equation 7 is the normalisation procedure on the raw weights $\mathbb{W}_{ijk}$, and Equation 8 shows the reconstruction method, which reverts k curves back to an AU sequence in mesh format.

As a result, given E_i and D_{ijk}, this branch method is able to compress meshes and an AU sequence into k curves (in the experiment, $k = 3$).

Mesh Curve Compression In Section 3.4.2, the weight curves W_i are clustered into $k = 3$ groups (clustering result in Fig 3-7). This branch method is a special case of Clustered Curve Compressing, when $k = 1$. In this branch, all the vertices in the mesh can be regarded as one group. Hence, by picking $k = 1$ and processing the equations in Section 3.4.2, the geometric information of an AU sequence can be represented by the combination of a single curve W_i, neutral

frame N_i and its Expression Difference Map E_i.

Evaluation

Table 3.1: *Curve Compression vs. Principle Component Analysis*

Method	AU1	AU4	AU5	AU6	AU7	AU10	AU12	AU13	AU15	AU16	AU18	AU20
PCA (D = 10)	0.603	0.681	0.353	0.566	**0.243**	0.677	0.279	**0.440**	0.348	0.423	0.810	0.729
VCC	**0.343**	**0.254**	**0.293**	**0.199**	0.302	**0.081**	**0.278**	0.806	**0.223**	**0.160**	**0.218**	**0.417**
CCC (k = 3)	0.354	0.279	0.323	0.212	0.372	0.086	0.313	0.816	0.249	0.174	0.241	0.439
CCC (k = 1)	0.365	0.297	0.360	0.234	0.432	0.090	0.364	0.857	0.272	0.206	0.265	0.518

PCA: Principle Component Analysis
VCC: Vertex Curve Compression
CCC: Clustered Curve Compression

In this subsection, the performance of the proposed approach is evaluated. First, the performance of these methods are quantitatively compared with PCA for several AU sequences. According to Table 3.1, Vertex Curve Compression (VCC) shows the best result against Clustered Curve Compression (CCC) and PCA. However, due to the mechanism of VCC, its compression percentage is only 33.3%, while PCA and CCC perform the dimensionality reduction from the original dimension (more than 10k) to n $<$ 10. Nevertheless, when subsequently comparing PCA (D = 10) and CCC (k = 3), CCC (k = 3) still shows significant advantage over PCA.

Then, the proposed method is measured against PCA in spatial-temporal terms, with twelve AU sequences in the model (Section 3.3) being chosen. After compressing and decompressing the data, the compression error of four methods, namely, PCA, Vertex Curve Compression (VCC), Clustered Curve Compression (CCC, k = 1) and CCC (k = 3) are compared. The comparison results are shown in Fig 3-8, with the blue line representing the error of PCA, whilst those of VCC, CCC (k = 1) and CCC (k = 3) are presented in red, green and black lines, respectively, whereas the x-axis in the figure represents the time line of frames of each AU sequence. All the curve methods can be seen as being highly spatially-temporally related, which is not the case for PCA.

Conclusion

In this section, a novel linear compression method has been demonstrated, which has been applied to compress a temporally related dataset, such as the dynamic 3D morphable model. Firstly, the mechanism of the proposed method was explained. Then, it was evaluated by comparing it with PCA in both quantitative and spatial-temporal aspects, with the curve method showing impressive superior

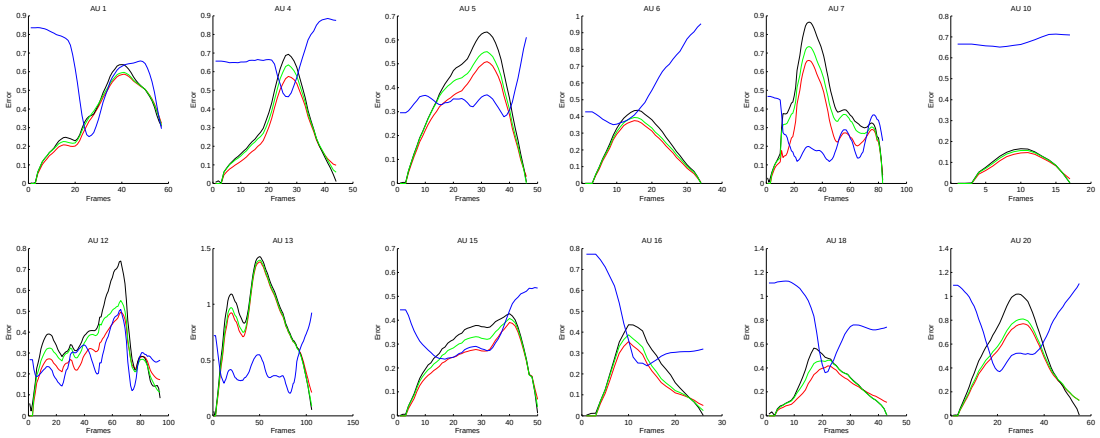


Figure 3-8: *RMS Curve Compression vs. PCA*

performance. To date, the curve method has only been applied to facial analysis. Applying and testing it in other temporally linked areas could be a fruitful direction for further research.

3.5 Transfer Facial Expression from 2D to 3D

3.5.1 Introduction

Over the past few decades, there has been an increased interest in automatic facial expression transfer techniques. In this section, I propose a novel facial retargeting technique pipeline for detecting and transferring the facial expression in 2D images to a 3D mesh for wild facial images in real time without any marker. This technique pipeline integrates the advantages of state-of-the-art methods of face detection, facial landmark detection and facial action unit recognition. Subsequently, the analysed results are applied to a template facial blendshape model in both the micro and macro levels. More specifically, in this section, I firstly discuss the previous research on facial landmark detection, facial action unit recognition and facial expression transfer. Then, details of the novel facial retargeting technique pipeline are introduced. Next, experimental results and evaluation of the technique pipeline are provided and discussed. Finally, a conclusion is given and potential further work discussed.

3.5.2 Previous Work

Facial Landmark Detection

As claimed by Zhang [166], facial landmark detection is a fundamental component in many face analysis tasks, such as facial attribute inference [104], face verification [102, 137], and face recognition [168, 169]. Whilst great strides have been made in this field [38, 39], robust facial landmark detection remains a formidable challenge in the presence of partial occlusion and the large number of head pose variations. Facial landmark detection is traditionally approached as a single and independent problem. Popular approaches include template fitting approaches [38, 167] and regression-based methods [39, 34]. Regarding which, Sun et al. [138] proposed detecting facial landmarks by coarse-to-fine regression using a cascade of deep convolutional neural networks (CNN). This method shows superior accuracy compared to previous ones [18, 34] and existing commercial systems. Nevertheless, the method requires the complex and unwieldy cascade architecture of a deep model.

Facial Action Unit Recognition

Facial Action Unit (AU) detection has been studied extensively in the last few years. The majority of the research works on this topic can be divided into AU occurrence detection and intensity estimation [146]. Additionally, Fasel [57] and Zeng's [164] survey on this research field provides a general overview of facial expression recognition.

AU occurrence detection Common binary classifiers applied to this problem include Artificial Neural Networks (ANN), boosting techniques, and Support Vector Machines (SVM). ANNs were the most popular method in earlier works (e.g. [15, 143]) and boosting algorithms, such as AdaBoost and GentleBoost, have since become a popular choice for AU recognition (e.g. [72, 159]). Boosting algorithms are simple and quick to train, they have fewer parameters than SVM or ANN as well as being less prone to overfitting. They implicitly perform feature selection, which is desirable for handling high-dimensional data and speeding up inference along with being able to handle multiclass classification. SVMs are currently the most popular choice (e.g. [36, 108]) as they provide good performance, can be non-linear, parameter optimisation is relatively easy as efficient implementations are readily available and a choice of kernel functions provides them with remarkable flexibility in terms of design.

AU intensity estimation The goal in AU intensity estimation,as claimed in [147], is to assign a perframe label with a possible integer value from 0 to 5 for each AU. This problem can be approached using either a classification or a regression learning method. Regarding the former, some approaches use the confidence of a (binary) frame-based AU activation classifier to estimate AU intensity. The rationale behind this is that the lower the intensity is, the harder the classification will be. For example, Bartlett et al. used the distance of the test sample to the SVM separating hyperplane [14], while Hamm and his colleagues used the confidence of the decision given by AdaBoost [72]. It is, however, more natural to treat the problem as a 6-class classification. For example, Mahoor et al. employed 6 one - vs.- all binary SVM classifiers [108]. Alternatively, a single multi-class classifier (e.g. an ANN or a Boosting variant) can be used. The extremely large class overlap means, however, that such approaches are unlikely to be optimal. In relation to regression-based methods, AU intensity estimation is nowadays often posed as a regression problem. These penalise incorrect labelling

proportionally to the difference between the ground truth and prediction. Such ordinal consideration of the labels is absent in classification methods. The large overlap between classes also implies an underlying continuous nature of intensity that regression techniques are better equipped to model. Examples include Support Vector Regression ([81] and [129]), whilst Kaltwang et al., instead, used Relevance Vector Regression to obtain a probabilistic prediction [84].

Facial Expression Transfer

Facial expression transfer refers to [141], also called facial motion retargeting or face cross-mapping, is the technique of adapting the motion of an actor to a target character. Nowadays, it is also highly pertinent to research areas, such as facial performance capture and performance-driven animation, having become very popular. Several approaches have been proposed for facial expression retargeting, aimed at transferring facial expressions captured from a real subject to a virtual Computer Graphic (CV) avatar [153, 29, 33]. Facial reenactment goes one step further by transferring the captured source expressions to a different, real actor, such that the new video shows the target actor reenacting the source expressions photo-realistically. Reenactment is a far more challenging task than expression retargeting, as even the slightest errors in the transferred expressions and appearance mean that such inconsistencies with the surrounding video will be noticed by a human user. Most methods for facial reenactment proposed so far involve working off-line and only few of the produced results are close to photo-realistic [43, 62].

3.5.3 Methodology

In this section, I outline the core technologies used in the technique pipeline for facial behaviour analysis and facial expression transfer. First, I discuss the details of how I detect and track facial landmarks. Then, a facial action unit intensity technique is introduced. Finally, a novel facial expression transfer algorithm is provided.

3.5.4 Facial Landmark Detection

In the technique frame, the OpenFace Toolkit [13] is used, an open source toolkit, for detecting the face and facial landmarks from 2D images. This toolkit implements a state-of-the-art facial landmark detection algorithm, which is an

Figure 3-9: *An example of facial landmark detection result*

upgraded version of Conditional Local Neural Fields (CLNF) [12] . More specifically, as an instance of a Constrained Local Model (CLM) [42], CLNF, however, uses more advanced patch experts and optimization function. Check meaning There are two main components of CLNF, the: Point Distribution Model (PDM) and patch expert [13]. The former captures landmark shape variations, whilst the latter capture local appearance variations of each landmark [12]. However, the originally proposed CLNF model performs the detection of all 68 facial landmarks together. Nevertheless, Baltrusaitis' model [13] shows more accurate and robust results by training separate sets of point distribution and patch expert models for eyes, lips and eyebrows. Figure 3-9 demonstrates an example of a facial landmark detection result by the using Openface Toolkit.

3.5.5 Action Unit Detection

In the proposed technique framework, similar to the implementation of facial landmark detection, OpenFace Toolkit [13] is adopted. The AU intensity detection module in the OpenFace Toolkit is developed based on a state-of-the-art framework [11, 145]. By utilising the OpenFace Toolkit, the intensity of action

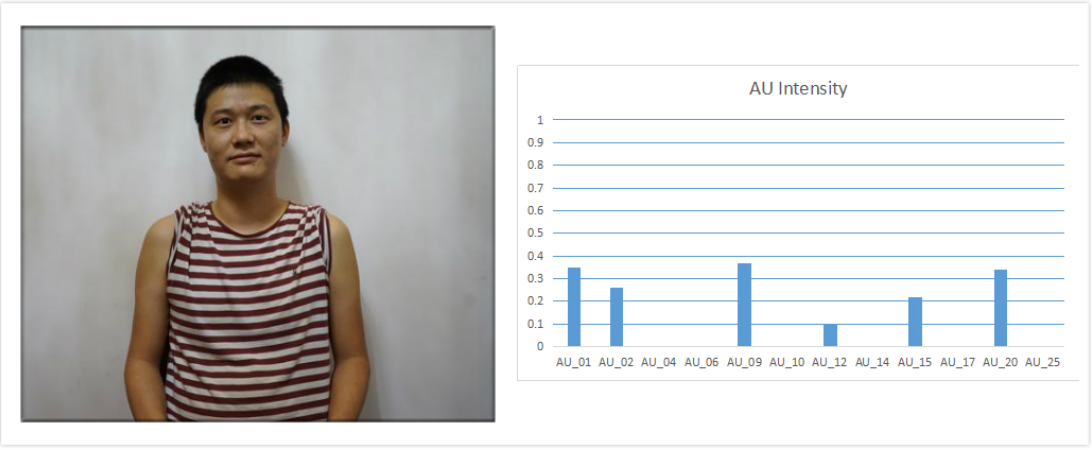


Figure 3-10: *An example estimation of facial action unit intensity detection*

units (actually, AU1, AU2, AU4, AU5, AU6, AU9, AU10, AU12, AU 17, AU20, AU25 and AU26) from faces in 2D images can be detected and measured. As shown in Figure 3-10, the intensity of the AUs detected by the Openface Toolkit and represented by positive float numbers.

3.5.6 Facial Expression Transfer

A novel facial expression transfer method is introduced in this section. After a substantial amount of experiments and testing, the OpenFace toolkit provided robust performance in both detecting facial landmarks and estimating the intensity of facial AUs [13] on wild facial images. Consequently, the advantages of OpenFace are combined here with the excellently designed facial expression transfer framework introduced by Ravikumar [120] in 2016. Ravikumar's system offers a template blendshape model (Figure 3-11) for the purpose of non-rigid deformation on the human face. In particular, there are over one hundred controllable blend attributes in the model, which allows scientists preciously apply the observed facial action unit onto the model in detail. Moreover, in order to fit the model correctly, a strategy for doing so in both the macro and micro levels is designed. In the macro level, the landmark detection result is used to control the deformation of the eyes and mouth. Then, at the micro level, the intensity estimation result is deployed to control the deformation of related blendshape atrributes of each AU. However, because of the fact that muscles that control the AUs on lower face are much more intensive than those on the upper face, the

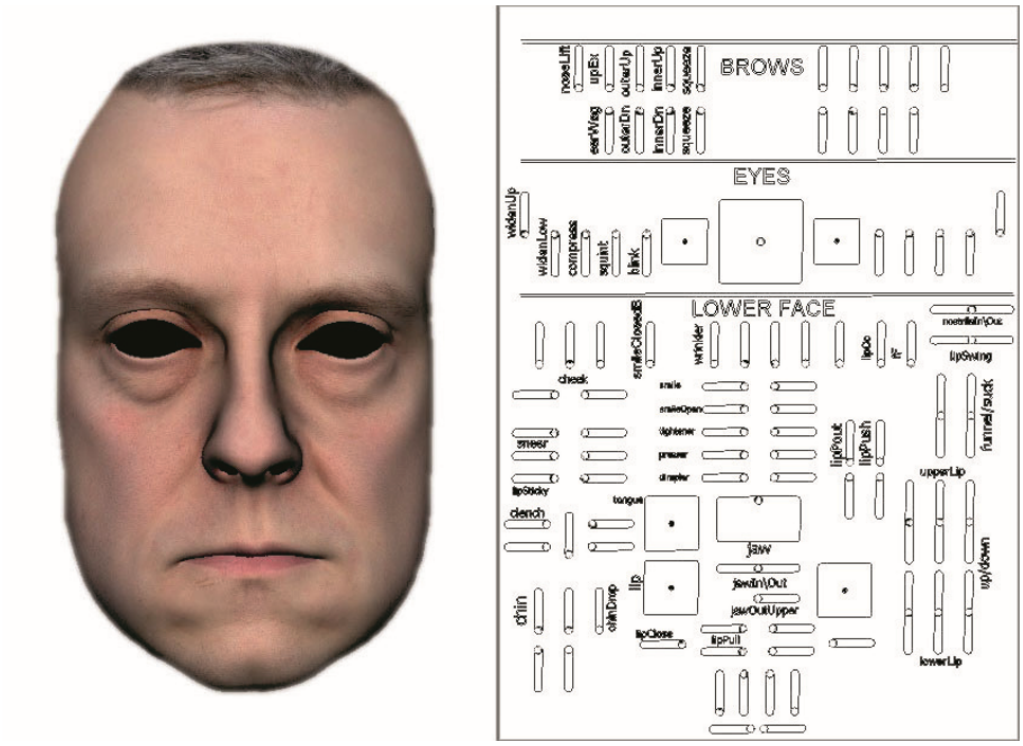


Figure 3-11: *Ravikumar's Facial Blendshape Model*

intensity estimation result of the AUs on the former is normally too strong. To accommodate for this, the intensity estimation result of lower faces are multiplied by 0.2 experientially.

3.5.7 Evaluation and Experiment Results

In this subsection, the results of facial expression transfer from 2D images to 3D models are presented and evaluated visually. More specifically, experiments are performed on males and females, separately and there is no marker on the subjects' faces. Figure 3-12 and Figure 3-13 demonstrate different facial expressions captured by a consumer-level camera in arbitrary lighting environments. The OpenFace Toolkit is, firstly, utilised to obtain facial landmarks and intensities of facial AUs in both the lower and upper face. Then, the novel facial expression transfer method is adopted in order to apply non-rigid deformation onto the target 3D face mesh. According to the visual results, the proposed method

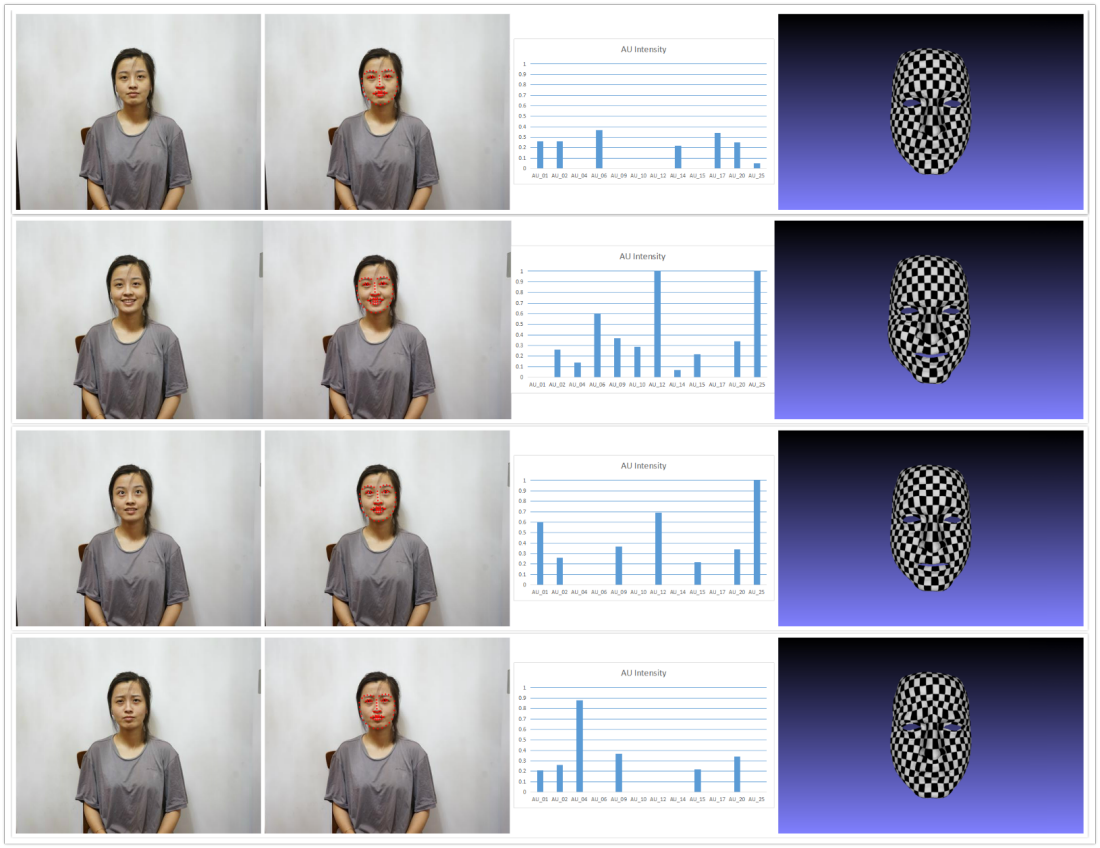


Figure 3-12: *Results of Facial Expression Transfer From a 2D image to a 3D mesh on a female. From left to right: original image, detected landmarks, estimation of AU intensity and facial expression transfer results on the 3D mesh*

provides accurate results in relation to the transfer of facial expressions from 2D images to the 3D mesh. This is because this macro level control of blendshape guarantees a high level of correspondence, whilst the micro level control provides the reference for adjusting the details of the AUs on the faces. Moreover, due to the performance of the OpenFace toolkit and the direct facial transfer algorithm, it takes only 40ms to finish the entire transfer process for each frame. However, as can be seen in Figure 3-12 and Figure 3-13, there is still some mismatching of the motion of muscles on the lower face as well as around the mouth.

3.5.8 Conclusion and Further Work

In this section, related work on facial landmark detection, facial action unit recognition and facial expression transfer was discussed. Then, a novel technique pipeline for transferring facial expression from 2D images to 3D meshes

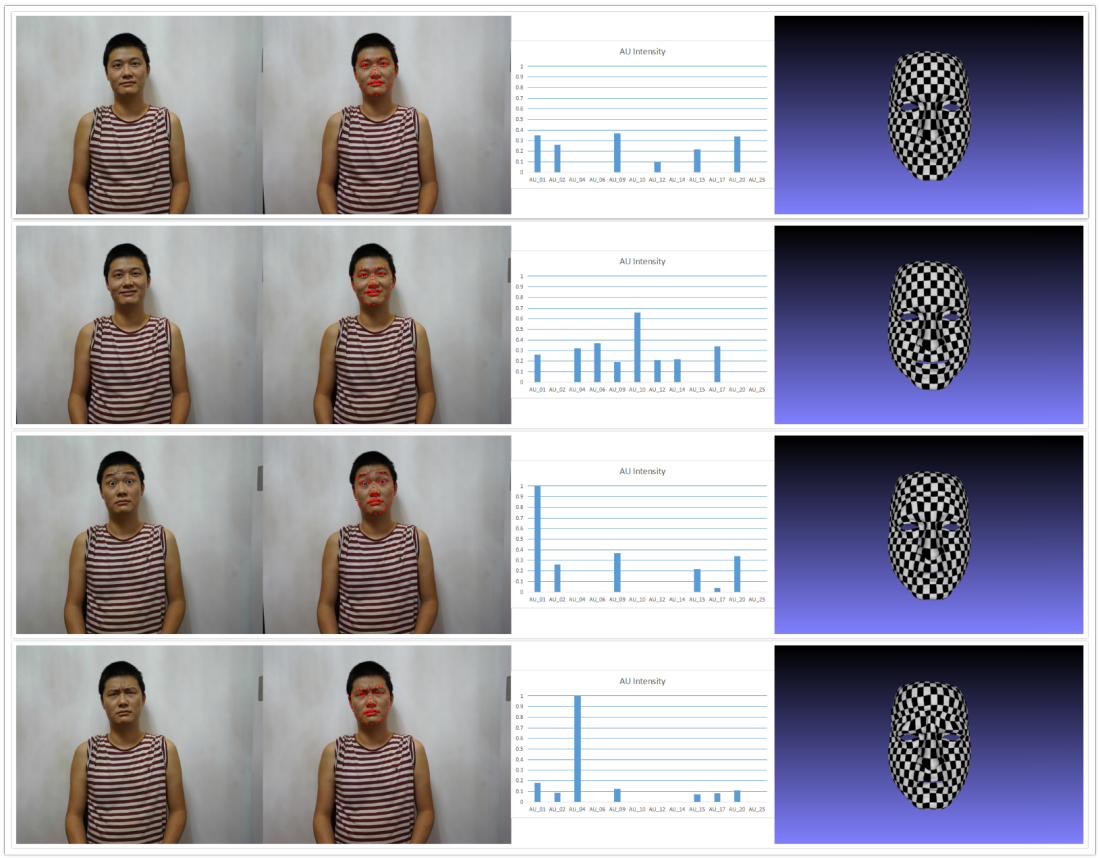


Figure 3-13: *Results of Facial Expression Transfer From 2D a image to a 3D mesh on a male. From left to right: original image, detected landmarks, estimation of AU intensity and facial expression transfer results on the 3D mesh*

was presented. Subsequently, some experimental results were demonstrated and there was evaluation of the technique pipeline. According to the visual evaluation, the proposed framework shows a robust real-time performance in relation to transferring expression from a 2D image to a 3D mesh for images without any facial markers. However, as mentioned before, some details of the motion of facial muscles still cannot be detected and apply onto 3D mesh. Hence, some further research on detecting specific areas of the face, in particular, around mouth, could well prove beneficial. Additionally, this work does not consider the pose variance of faces. Therefore, another potential work is to add pose estimation on 2D images and apply the estimated pose on to 3D mesh.

3.6 Conclusion

Facial analysis in 3D has become a salient topic in the computer vision and graphics community. Compared with 2D face models, dynamic 3D face models are more robust towards pose and light variance. That is, they deliver significant advantages when compared to static face models in analysis, such as face identity and facial expression recognition. In this chapter, the history of facial analysis has been discussed along with explanations being provided regarding the techniques for constructing a dense corresponded dynamic 3D model. Subsequently, the problems faced and potential solutions have been covered. In order to establish such a model, it is essential to develop accurate dense correspondence estimations of the vertices between face meshes. In this context, a novel non-rigid registration method has been proposed, which is capable of processing and constructing a dense corresponded dynamic 3D facial dataset. Furthermore, this method has been evaluated by comparing it with the-state-of-art approach. Then, in order to compress and model the dense corresponded dynamic 3D dataset, a novel linear compression method was presented, in which the compression result is highly spatially-temporally related. In the evaluation, the algorithm was tested against PCA in both quantitative and temporal terms. At the end of the chapter, some potential further 3D modelling work has been put forward.

CHAPTER 4

AUTOMATIC CAD FLOOR PLAN REGULARIZATION AND COMPONENT EXTRACTION SYSTEM

Summary

This chapter presents an automatic system for the analysis and labelling of floor plan drawings in computer-aided Design (CAD) format. The proposed system applies a fusion strategy to detect and recognise various components of CAD floor plans, such as walls, doors and windows. Technically, a general rule-based filter parsing method is fist adopted to extract effective information from the original floor plan. Then, an image-processing based recovery method is employed to correct information extracted in the first step. Our proposed method is fully automatic and real-time. Such analysis system provides high accuracy and is also evaluated on a public website that, on average, archives more than ten thousand effective uses per day and reaches a relatively high satisfaction rate.

4.1 Introduction

CAD floor plans [32] comprise a set of architectural drawings that describe the layout of various structural objects (e.g. walls, windows, doors and furniture) in a building.

In architecture and building design, floor plans contain various levels of detail and show the relationships among rooms, spaces and other architecture components for each level of a structure.

Floor plan analysis can be considered a special image analysis method that attempts to understand the structural and semantic information of a building by analysing 2D versions (in this chapter,'images' refers to both rasterised and vectorised images). After reviewing previous research, it is clear that there are various purposes for analysing a given floor plan. For example, several studies have applied floor plan analysis to the generation of 3D models [49], [103], [113], and another study emphasised interpreting floor plans as CAD formats. In addition, other studies have attempted to detect rooms in architectural drawings [107], [154] and have also searched massive floor plans [152].

[10] is similar to [99] in that they both proposed a method to understand hand-drawn floor plans. In addition, a previous study proposed a complete system for architectural diagram analysis [50], where basic primitives are recognised by applying numerous automated graphics-recognition processes. A method to detect rooms in architectural floor plan images has also been proposed [107]. That method was then adopted and expanded [5] to include new processing steps, such as wall edge extraction and boundary detection.

This chapter presents an automatic system for analysing floor plan drawings in CAD format. The remainder of this chapter is organised as follows. Work related to this chapter is summarised in Section 4.2. Section 4.3 provides an overview of the proposed method, including its specific processing steps. Section 4.4 presents an evaluation of the proposed analysis method and discusses experimental results. Finally, Section 4.5 concludes this chapter and offers suggestions for future work.

Figure 4-1: *Different ways to draw a wall with a window and a door. The variable graphic symbols pose challenges for automatic recognition of objects in CAD drawings. [163]*

4.2 Related Work

Architectural drawings, typically in the form of floor plans, are necessary to design, describe and execute a construction project. The architectural elements on each building level are represented using standard symbols and floor plans typically create a top-down orthographic projection.

Floor plans consist of various levels of detailed architectural elements. For example, construction structure drawings (CSD), which are one of the most complicated types of floor plan, portray internal steel bars, the concrete structure of columns, beams, WA walls as well as pipe and ductwork layouts. These drawings are popular with both design engineers and construction managers. Tong Lu and his research team [103] introduced a system that constructs a detailed building model from computer-drawn CSDs. However, interpreting raster images of CSDs requires further research.

Despite floor plans being widely used in architecture engineering and the construction life-cycle as they are able to cover a building's complete layout, both hand-drawn and computer-produced forms often lack detailed construction information.

Another main drawback arises from the various graphical symbols used in floor plans. Figure 4-1 shows several common graphical symbols for walls, windows and doors. Note that not all floor plan drawings comply with specific standards. However, the overall purpose of floor plan drawings determines which ones and how components will be shown. The various symbols create a challenge when analysing and interpreting floor plan drawings, especially for shape analysis or shape matching methods.

4.2.1 Converting Floor Plan CAD files

Systems that apply CAD-based floor plans focus more on 3D model extrusion rather than image processing and pattern recognition. Rick Lewis and Carlo Sequin [92] at the University of California, Berkeley, introduced a system that creates 3D polygonal building models semi-automatically by grouping architectural symbols into specified layers in standard DXF files. Their system introduced a correction strategy on disjointed and overlapping edges in order to overcome geometric flaws. This system collects the topology of spaces and portals to generate proper polygon orientation. After each floor is modelled, the system stacks the floors to create a complete model. This system significantly simplifies the recognition process, which benefits designers in various applications, such as smoke propagation simulations.

Clifford So [135] and his colleagues from the Hong Kong University of Science and Technology (HKUST) considered the model conversion problem in a virtual reality context. They targeted three major tasks, i.e. wall extrusion, object mapping as well as ceiling and floor contraction, after observing model reconstruction via a conventional manual method. The processing time of their method is greatly reduced by incorporating automated approaches for each task in the next step, including automatic wall polygon extrusion, generating and placing customised templates of random orientation and size as well as advancing front triangulation. However, their system has a significant disadvantage, i.e. the input file must contain fully established semantic information and no errors, which means the system requires manual intervention. For example, wall lines must be marked by users, architectural objects must be specified, and objects must be assigned to individual transformation matrices.

Researchers at the Massachusetts Institute of Technology (MIT) [26] automated the construction of a realistic MIT campus model (Building Model Generation project (http://city.csail.mit.edu/bmg). Compared to the Berkeley system [92] and whilst a similar pipeline is employed, an additional process is used to position and orient building models automatically using a map for guidance.

Lu's research team at the Nanjing University of China proposed a system to construct models from computer-drawn CSDs and vectorised floor plans [103]. Compared to other computer drawing formats, symbol recognition in a vector image is much more difficult, because such images contain unlabelled geometric primitives. Similar to the HKUST project [135], this system differentiates walls from other architectural elements. First, it detects parallel line segment pairs as

Table 4.1: *The challenges of image parsing and drawing analysis*

Step	Issues
Noise removal	Notation leading lines can easily be confused with wall lines. The background may contain a grid or decorative pattern.
Text extraction	Textfont, size and orientation may vary. Text and graphical symbols may share pixels (overlapping or touching).
Vectorisation	Most algorithms recover only lines and arcs. Free-form curves are a challenge. Noise affects the result significantly. Vectorisation may yield poor results at junction points.
Symbol recognition	Symbols may not comply with standards. There may be a large pool of symbols, and the differences between two symbols may be subtle.

walls, which are then removed from the drawings. Next, the remaining primitives are recognised by detecting feature matches with predefined patterns that contain a symbol's graphical primitives and contextual information. In the recognition process, the system places patterns in order relative to their priority and checks each, one by one. Corresponding elements are removed from the drawing as soon as they satisfy all of a given pattern's constraints. Whilst this does require high-quality input, the system benefits users significantly because it focuses on structural details and is highly automated.

4.2.2 Image Parsing and Drawing Analysis

This process analyses an input raster floor plan image and extracts layout information, which is referred to as a 'parse process'. Referring to Yin's survey [163], the challenges in this step are explained in Table 4.1.

Graphical document analysis technology is required to analyse and parse image floor plans, which includes two main steps: (1) removing noise, such as text and annotation; and (2) graphical symbol recognition. The cleaning step focuses on removing noise and other irrelevant information to improve image quality. In the graphical symbol recognition step, the system categorises the recognised symbols by identifying certain information, including location, orientation and scale.

Compared to other graphical documents, floor plans have certain distinguishable features. For example, various line shapes (curved or straight) represent walls in floor plans. Another difference is that the architectural symbols are made up of simple geometric primitives. Typically, to handle such input, graphics recognition is integrated with vectorisation.

(Table 4.1)

Noise Removal

Sampling noise introduced by digital scanning is very common when processing hand-drawn floor plans. However, they are generated by a computer gradually and thus, noise has a broader definition in this context. For example, pixels without directly useful information are typically considered noise, including annotation leading lines, dimension lines, furniture and hardware symbols. On rare occasions, a decorative pattern in the background can be misidentified.

In Loria's system [101], a morphological filter is applied as a fine line between noise and useful pixels. This method is based on the assumption that background patterns and dimension leading lines can be differentiated from useful lines, because they have different thicknesses and styles. [113] makes a similar assumption, filtering input and only thick construction lines can be preserved.

Text Extraction

A perfect algorithm should be free from text font, size and orientation as well as being efficient and requiring little manual intervention. Geometric shapes mixed with text incur extra burden for separation and extraction tasks. Text research has been developed for several decades, and its results can be categorised into structural-based (focusing on structural differences) and pixel-based algorithms.

Graphic Recognition

The text is separated from graphics in the previous step. Graphic recognition is a process whereby pixels are organised and ordered according to the geometrical description of the building's layout. Typically, architectural drawings comprise two primary types of information, i.e. structural information and local architectural components.

As shown in Table 4.1, graphic recognition comprises vectorisation and symbol recognition. Walls are preserved as geometric poly-lines for the extrusion step, because they define the building's spatial structure. From this perspective, all systems introduce vectorisation and deal with geometric elements, rather than performing symbol recognition on pixels directly.

Vectorisation This process, which is referred to as raster-to-vector conversion, transfers image pixels to geometric primitives. The most important aspects of

each algorithm are efficiency, robustness and accuracy. The workflow of traditional line-drawing vectorisation involves two steps, as shown in the following table.

It should be noted that correcting joint errors is required after each step. In most cases, vectorisation algorithms can find line segments and circular arcs; however, more complex curves remain a challenge for existing algorithms.

In Step 1, three groups of algorithms, i.e. parametric model fitting, contour tracking and skeletonisation [77], are typically used. In parametric model fitting, Hough transform [51] is applied to detect lines; however, this requires significant amounts of memory and lacks universality.

Contour tracking detects the contour of white pixels (rather than black ones) and recognises connected regions as rooms. This method can deal with simple floors; however, it cannot deal with complicated structures, because it is based on the assumption that white spaces are divided by black wall lines in the image.

Thinning-based algorithms for skeletonisation attempt to search for a curve bones' medial axis by stripping boundary pixels until a one-pixel wide skeleton remains [90] . Here, one disadvantage is that intersections always confuse the results. Another disadvantage is that thinning-based algorithms require significant time to process, because each pixel is visited multiple times. Typical medial-axis-based algorithms include pixel tracking [48] and run-graph-based algorithms [48]. Medial-axis-based algorithms treat a thick line as a solid shape and its medial axis as a skeleton.

In Step 2, point chains are segmented into sets of lines, poly-lines and circular arcs by estimating curvature or polygonal approximation to identify critical points.

Loria's system introduces a skeletonisation technique and polygonal approximation to complete the vectorisation process [101]. The CUHK system tracks the contour of black pixels rather than white ones, which differs from contour tracking [113].

Symbol Recognition This is the most important part of graphical document analysis, and the graphic symbol recogniser (GSR) in this process should be efficient and not limited to either context or affine transformation. It should be noted that previous research has proved that several methods work well in specific areas and generate positive results.

GSRs can be classified as vector based (oriented toward structure) and pixel

based (oriented toward statistics). Vector-based GSRs process graphical primitives, such as points, line segments, arcs and circles, in vectorised images. This approach checks primitives in groups to identify a symbol using region adjacency graphs [4], graphical-knowledge-guided reasoning [158], constraint networks [4] and deformable templates [148]. Note that good vectorisation is expected with this approach, which is affine invariant.

Other GSRs are pixel based. Such recognisers process raster images without vectorisation. Such methods focus on the statistical features of a symbol's pixel information. Pixel-based approaches contain plain binary images [130] , living projections and shape contexts [19] . Compared to vector-based approaches, pixel-based ones are more accurate, even though their performance is sensitive to scaling and rotation. Su Yang improved a recognition method by merging pixel-based and vector-based approaches [160].

In Loria's project, a network is applied to identify the features of a vectorised image's primitives [101]. Then, segments in the vectorised floor plan are distributed throughout the network to find terminal symbols. A similar, but simpler approach, is employed in the CUHK system, in which a series of geometric constraints are considered symbol patterns. With this approach, raster or vector images of a floor plan can be used to improve recognition accuracy [113].

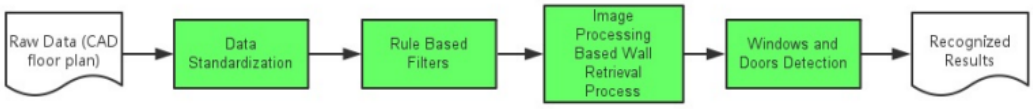


Figure 4-2: *Work flow of the floor plan analysis system. Starting with putting in raw data, followed by the process of standardisation, filtering and rasterising correcting. As a result, walls, windows and doors can be detected*

4.3 Floor Plan Analysis System

The proposed automatic floor plan analysis system targets engineering uses and takes CAD format floor plans, which can be considered a set of vectorised images with unit information, as input.

Figure 4-2 illustrates the basic workflow of the proposed floor plan analysis system, which is described in detail in the following. As mentioned previously, the proposed system is available online for engineering uses without any restrictions to the drawing style of a floor plan. This means that the proposed system can accept a wide variety of input data. Therefore, standardised parsing is required to normalise input data in the first step. After standardisation, all of the information provided by the floor plans is represented by the most basic elements, i.e. straight lines and arcs. Then, because floor plans represent structural data, such as walls, windows and doors, a general filter is applied to the lines to obtain effective room structure information. Then, in consideration of excessive filtration in the filtering step, an image-processing-based retrieval method is adopted to correct the filtered result. Finally, the proposed is able to system extract windows and doors from the floor plan.

4.3.1 Data Standardization

Problem Statement

It is difficult to develop a general automatic system for recognising various types of CAD floor plans, because designers and engineers draw them in different ways. The variety of CAD floor plans can be summarised as follows.

(1) Different units are adopted in different CAD floor plans. Architecture designers employ different units (e.g. centimetres, inches and millimetres), according to the given project's requirements or personal preference.

(2) Structural objects can comprise various internal forms. For example, as shown in Figure 4-1, doors and windows can be drawn in different ways [163] , and, as shown in Figure 2.b, even in a single CAD floor plan, load bearing walls (filled polygons) differ from normal walls (parallel lines). Such variable graphic symbols pose challenges when attempting to recognise floor plans automatically, for example, shape matching.

(3) Dimensions are varied in CAD floor plans, according to the intended purpose. For example, for architectural purposes, some CAD floor plans are shown in 3D space, while others are shown in 2D space.

(4) In most cases, manifold furniture or annotations are applied, which impede recognition of the primary structural components (i.e. walls, windows and doors). In most cases, manifold furniture or annotations are applied, which greatly disturb the recognition of the main structural components (walls, windows and doors).

(5) Some CAD drawings may contain several floor plans in a single drawing (Figure 3), with each one in such drawings being independent. Hence, the proposed system must be able to extract and separate the individual plans.

Solutions

A simple data standardisation process is employed to address the variety of input CAD floor plans. The primary purpose of this process is to normalise all the architecture elements in the floor plans as lines. The process is described as follows.

a. The first step is to standardise the units. By reading the unit information of the CAD floor plans, the various units are converted to millimetres.

b. Regardless of the composition of structural objects (e.g. lines, solids, triangles, multi-lines, poly-lines or blocks), all such objects in the drawing are decomposed into lines, which is the most basic element in all architecture drawings. Simultaneously, arcs are converted to short and continuous lines.

c. The proposed system focuses on detecting walls, doors and windows in a 2D CAD floor plan and hence, 3D floor plans are converted into 2D spaces by calculating the normal of the lines.

d. Furniture, which is considered a type of structural object, is decomposed into lines (refer to Step b). Regarding annotations, marker lines are converted into straight lines, and text elements are removed.

e. At this point, the architectural drawing now comprises straight lines and

Figure 4-3: *An example of multiple floor plans in a single CAD drawing; systematic clustering is employed to classify lines based on Euler distance*

arcs. Since the input drawings can contain more than one floor plan, systematic clustering of the lines is employed, based on Euler distance. I define the distance between lines by searching the closest link between lines (Figure 4-3).

f. Then, a 5000-mm threshold is applied so as to cluster all the lines to segment multiple floor plans from a single CAD file.

4.3.2 A Fusion Strategy System for CAD Floor Plans Analysis

Here, I introduce a fusion strategy system for CAD floor plan analysis. Floor plans always contain information that helps an architect express the actual layout of the structural objects, e.g. walls, doors and windows. However, during floor plan analysis, different types of objects must be interpreted at different points in time using specific strategies. Hence, a fusion strategy system is introduced that combines a set of general filters and an image-parsing method to extract walls, windows and doors from a CAD floor plan. In the filtering stage, the aim is to identify as many correct walls in the floor plan as possible. Walls are one of the essential elements in a floor plan,for other architecture components, e.g.

doors and windows, are attached to them. A set of filters needs to be designed to extract information about walls from the input data. Then, by rasterising the CAD floor plan, the aim is to restore any walls that were filtered excessively in the image-processing strategy. Based on the wall analysis results, the proposed system seeks to detect windows and doors. Then, an image-processing-based wall restoration system is employed. Finally, a mechanism is used to detect doors and windows based on the detected walls. These processes are explained next.

General Filters

Similar to existing work [50] and [113], in this step, the proposed system extracts walls from a floor plan by applying general filters. The filters are built on the assumption that walls are represented by parallel lines in Figure 4-4. Based on this assumption, the filters search for parallel lines and define them as wall candidates. The workflow of the filters is explained below.

1. Gradient Filter ($pi/12$) The objective of introducing this filter is to find non-vertical and non-horizontal lines, because based on the assumption that walls lie horizontally and vertically in a floor plan, they can be targeted by applying this filter. It should be noted that some designers draw lines at a slight tilt, so the threshold is set to $pi/12$. In Equation 4.1, L_{raw} is the raw input line from the CAD floor plan, and L_1 represents filtered lines after applying the gradient filter. Then, the filtered lines are divided into two sets: the horizontal set (H_{s1}) and the vertical set (V_{s1}), as expressed by Equation 4.2 .

$$L_1 = f_{Gradient_{filter}(pi/12)}(L_{std}) \tag{4.1}$$

$$(H_{s1}, V_{s1}) = f_{splitHV}(L_1) \tag{4.2}$$

Figure 4-5 shows the filtered result L_1 after the gradient filter (threshold $pi/12$) is processed, while the orange and blue lines represent the horizontal set H_{s1} and vertical set V_{s1}, respectively.

2. Length Filter (2mm) As mentioned in Section 4.3.1, arcs are converted into short lines; however, this may generate many short lines. In consideration of the negative effects brought by irrelevant short lines, a length filter (Equation 4.3) is applied to eliminate interference by them, thus addressing this issue. The

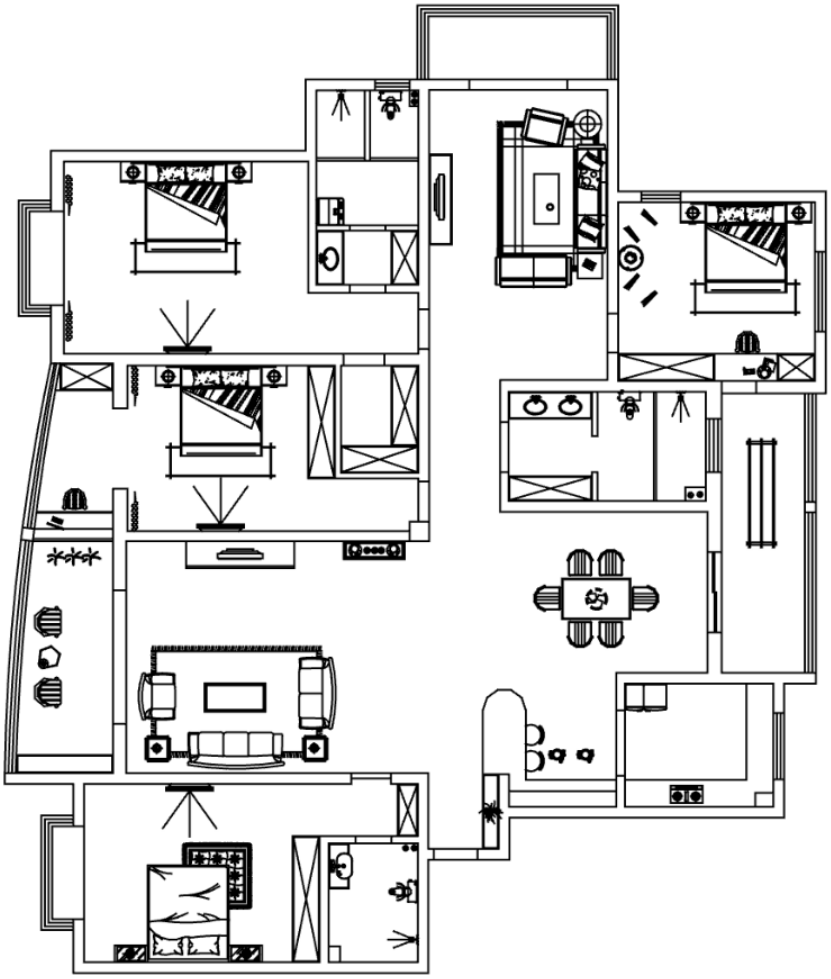


Figure 4-4: *Raw data as input of a floor plan. Parallel lines are targeted in the process of filtering, based on the assumption that they represent walls*

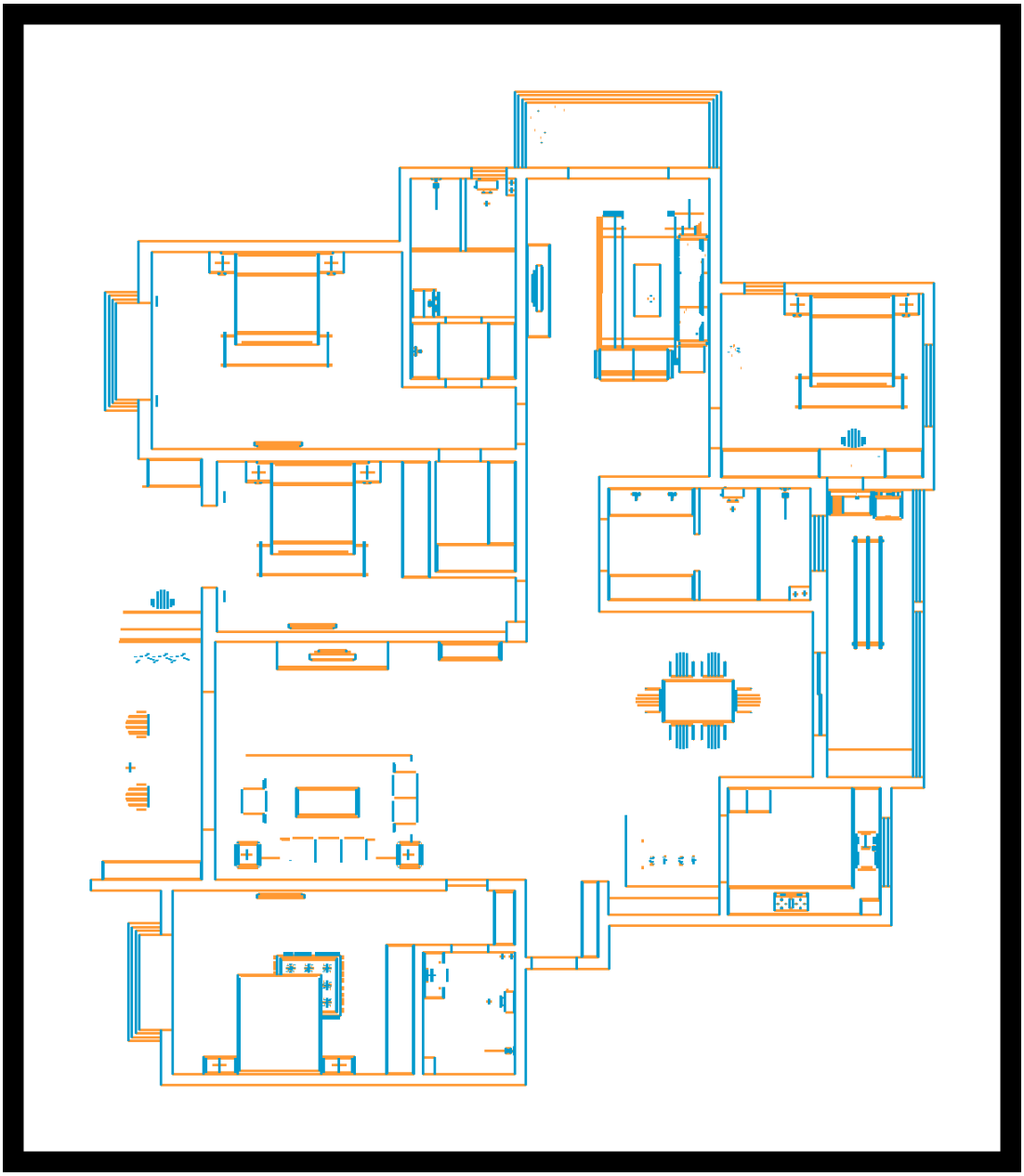


Figure 4-5: *Production of a gradient filter, with the orange and blue lines representing horizontal filtered lines and vertical set, respectively.(Threshold: pi/12)*

threshold of this length filter is set as 2mm in order to get the most accurate results and to improve work efficiency as well.

$$H_{s2} = f_{length_{filter}(1mm)}(H_{s1}), V_{s2} = f_{length_{filter}(1mm)}(V_{s1}) \tag{4.3}$$

In Figure 4-6, the red and blue lines represent the filtering results H_{s2} and V_{s2} after the length filter is applied, respectively.

3. Gap-Filling and Line Merging (1 mm, 1 mm, loop=5) In architectural drawings, small gaps or dislocations may be created when designers draw walls. The proposed system employs a gap-filling loop filter and merges close parallel lines to solve this problem. The gap-filling process is aimed at connecting close lines in H_{s2} and V_{s2} .

In this process, it is key to determine whether such lines are sufficiently close to each other, so the threshold is set to 1 mm. Then, the line-merging process merges lines in H_{s2} and V_{s2} are in a specific distance.Similar to the previous stage, a threshold of the same value (1 mm) is employed in this process in order to merge close parallel lines. Fig 4-7 shows the results obtained after applying the gap-filling and line-merging processes. In Equation 4.4, H_{s3} and V_{s3} are the filtered products of this process. It should be noted that this process is prone to drift errors; however, small errors will be fixed Section 4.3.2.

$$(H_{s3}, V_{s3}) = f_{merge(1mm)}(f_{fill(1mm)}(H_{s2}, V_{s2})) \tag{4.4}$$

4. Removing Multiple Parallel Lines Commonly, sets of close multiple parallel lines in walls with an equal gap size represent windows (Figure 4-8). This filter converts such multiple parallel lines in H_{s3} and V_{s3} into a pair of parallel lines. As shown in Figure 4-8 , if the outer bounds of such multiple parallel lines are connected to wall lines, a line-splitting filter is employed to split such long lines into segmented short lines.

$$(H_{s4}, V_{s4}) = f_{RML_{filter}}(H_{s3}, V_{s3}) \tag{4.5}$$

In Equation 4.5, H_{s4} and V_{s4} are the line-splitting filters.

After the line-splitting filter is applied, inner lines in the multiple parallel lines structure are removed and such structures must be detected in the floor plan to achieve this. To this end, a multiple parallel lines structure detector for H_{s3} and

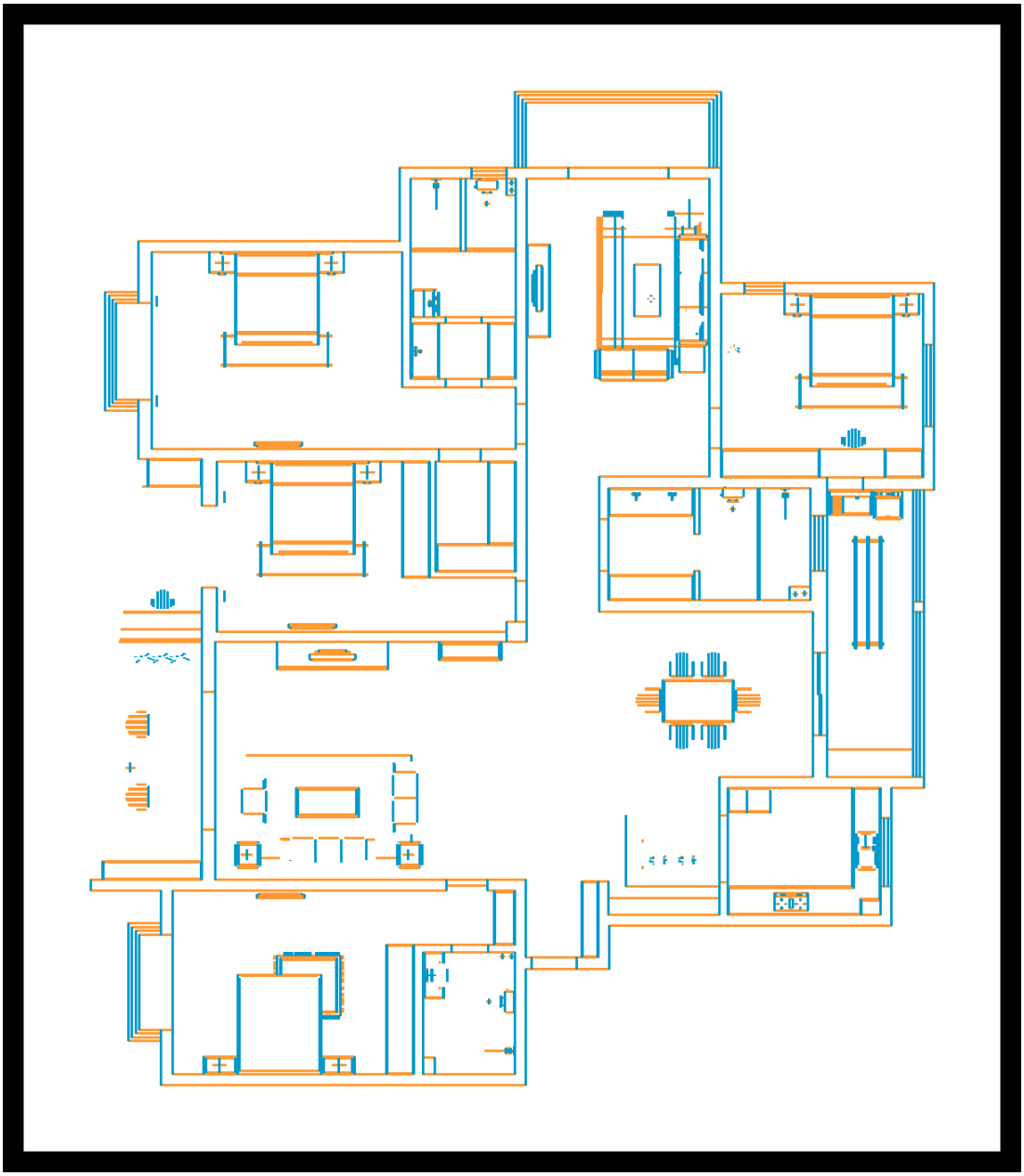


Figure 4-6: *Production of a length filter, with the red and blue lines representing the filtering result after it is applied (Threshold: 2mm)*

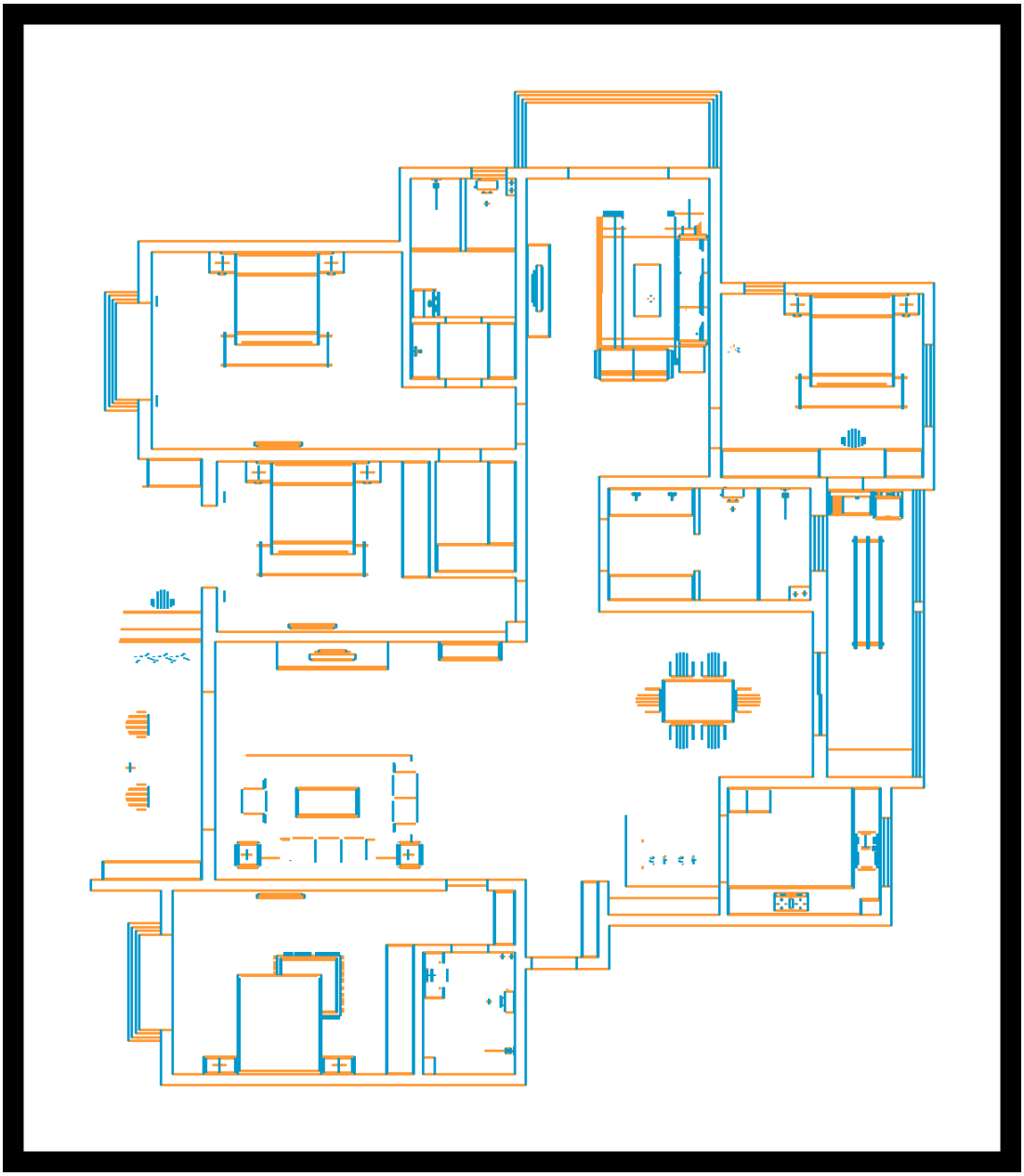


Figure 4-7: *Production of fill gap and merge lines processing. The gap filling process applies to lines that are within 1mm of each other*

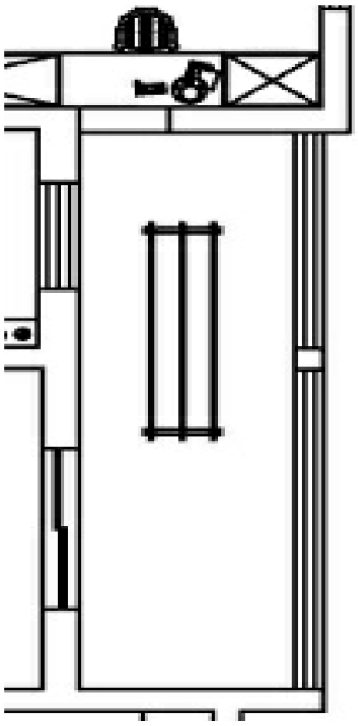


Figure 4-8: *Multi-parallel lines with same gaps to represent windows. Applying a line split function to split long lines into segmented short lines, because the outer bounds of windows are connected in walls*

V_{s3} is employed. For example, with H_{s3} ,the distance between lines is less than 300 mm and is marked as a benchmark, and lines in this range are placed into a candidate line group. It should be noted that inner lines in the candidate group are removed, if the number of multiple parallel lines is from three to six and the distance between each line is between 10 mm and 100 mm. Figure 4-9 shows the results obtained after removing such multiple parallel lines.

5. Length Filter (90mm) After removing the multiple-parallel lines, the remaining parallel lines in H_{s4} and V_{s4} can be considered as candidate lines for the construction of walls. However, many irrelevant lines that do not contribute to walls can remain in sets H_{s4} and V_{s4} .

As described in the previous subsection, the proposed system seeks to find as many correct walls as possible; however, some may have been over-filtered. The method used to restore walls is discussed in Section 4.3.2 .

Considering that main walls are typically represented by long pairs of parallel lines, a length filter with a threshold of 90 mm is employed to remove short lines that may cause interference. In Equation 4.6 , H_{s5} and V_{s5} denote the production of this length filter. Figure 4-10 shows the results obtained after applying it. The length filter removes many pairs of short parallel lines that construct short walls.

$$(H_{s5}, V_{s5}) = f_{length_{filter}(90mm)}(H_{s4}, V_{s4}) \tag{4.6}$$

6. Connectivity Filter In the proposed method, a line connectivity filter is applied relative to wall continuity. First, line connectivity is determined and then, lines that have no connection with any others are removed. Figure 4-11 shows the results obtained after applying the connectivity filter. In Equation 4.7, H_{s6} and V_{s6} denote the productions of the connectivity filter in the vertical and horizontal directions, respectively.

$$(H_{s6}, V_{s6}) = f_{connectivity_{filter}}(H_{s5}, V_{s5}) \tag{4.7}$$

7. Gap-Filling and Line Merging (90mm,50mm,loop = 5) For gaps between doors and long lines generated by placing furniture against walls, a gap-filling and line-merging filter is applied similar to that discussed in Section 4.3.2 . In Equation 4.9, H_{s7} and V_{s7} denote the productions of the gap-filling filter and

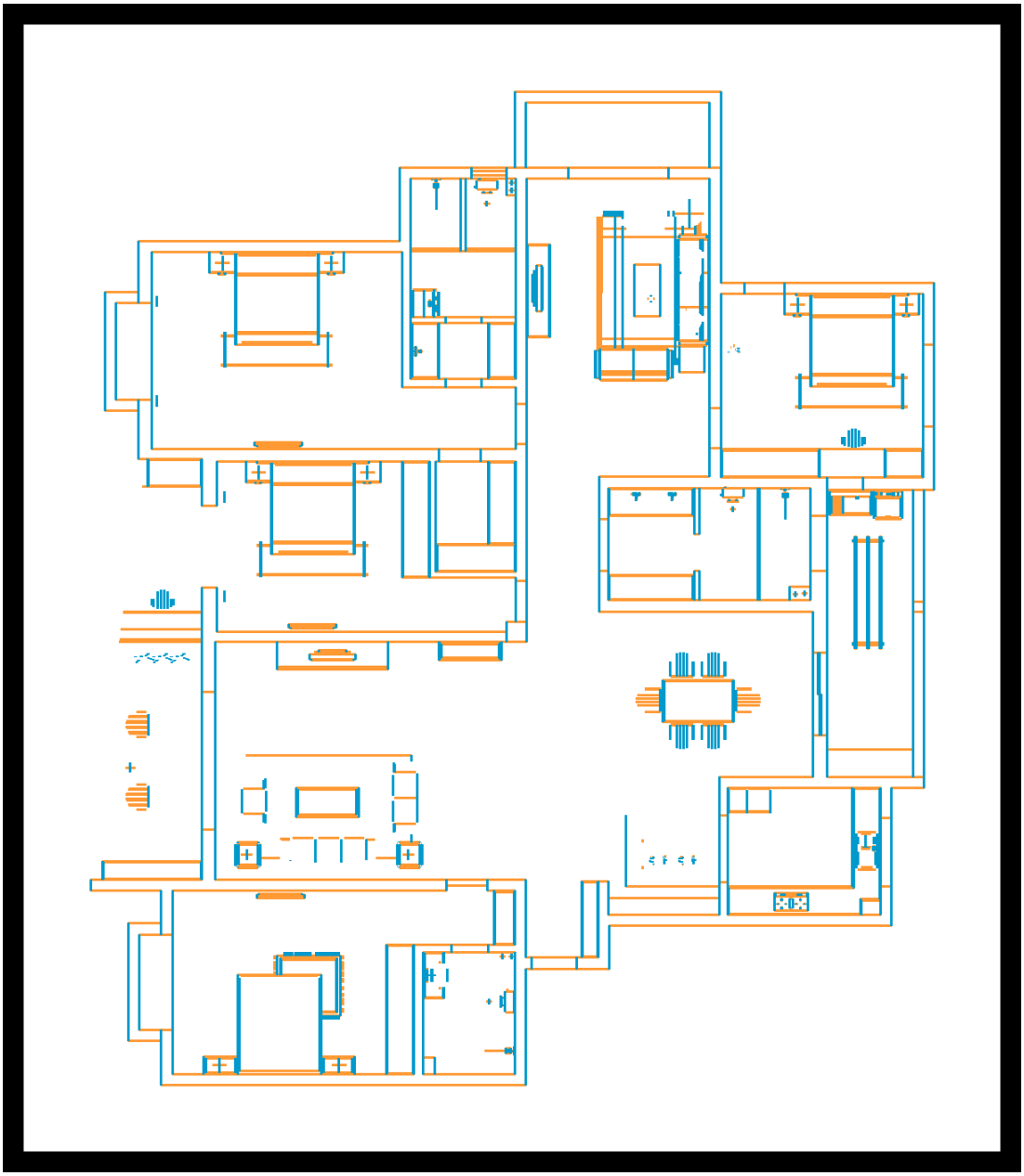


Figure 4-9: *The results after removing multiple-parallel lines. Inner lines in the multiple parallel lines structure are removed after applying a line-splitting filter*

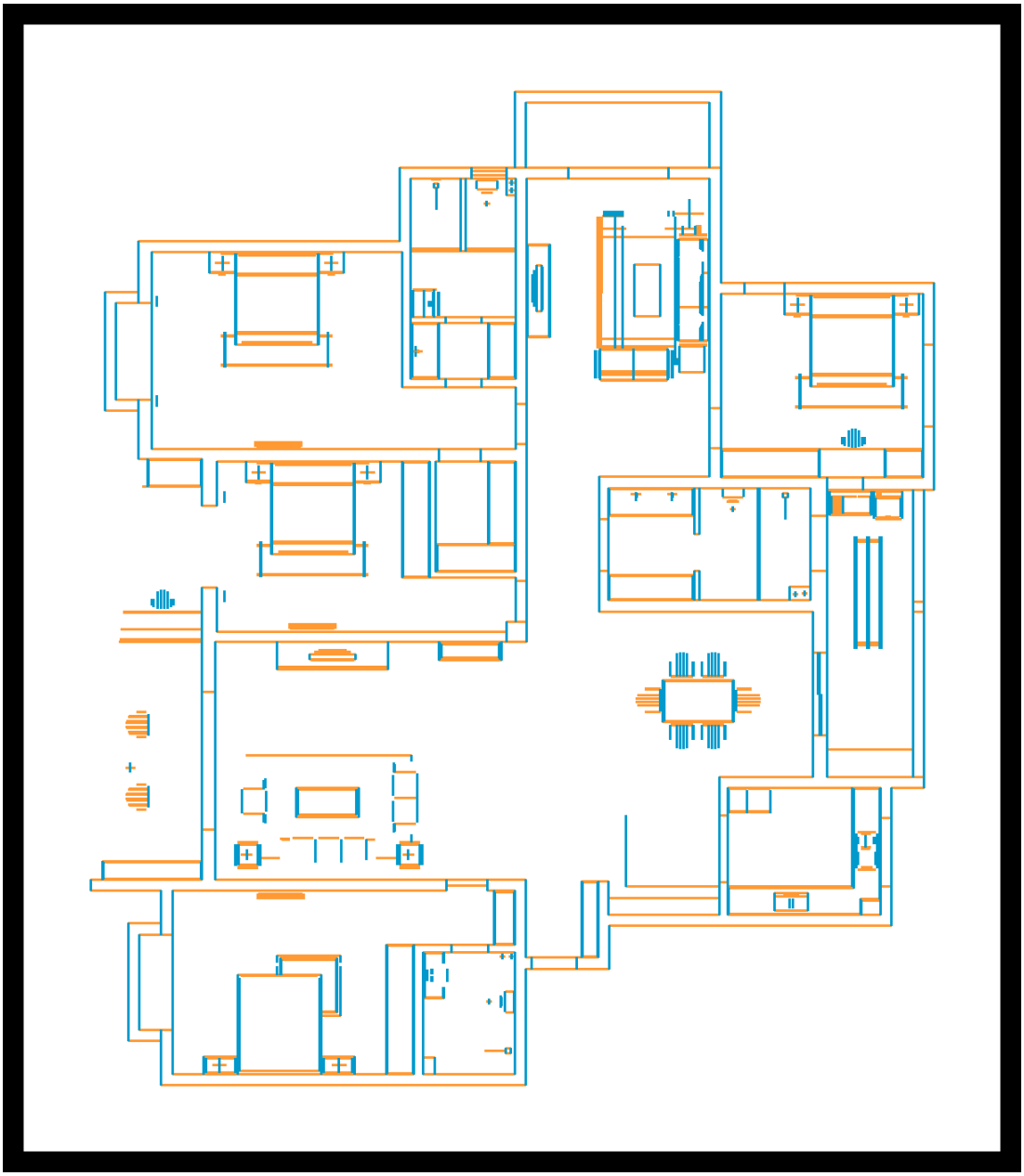


Figure 4-10: *The results after applying the length filter. It removes pairs of short parallel lines (less than 90mm) that contribute to a short wall*

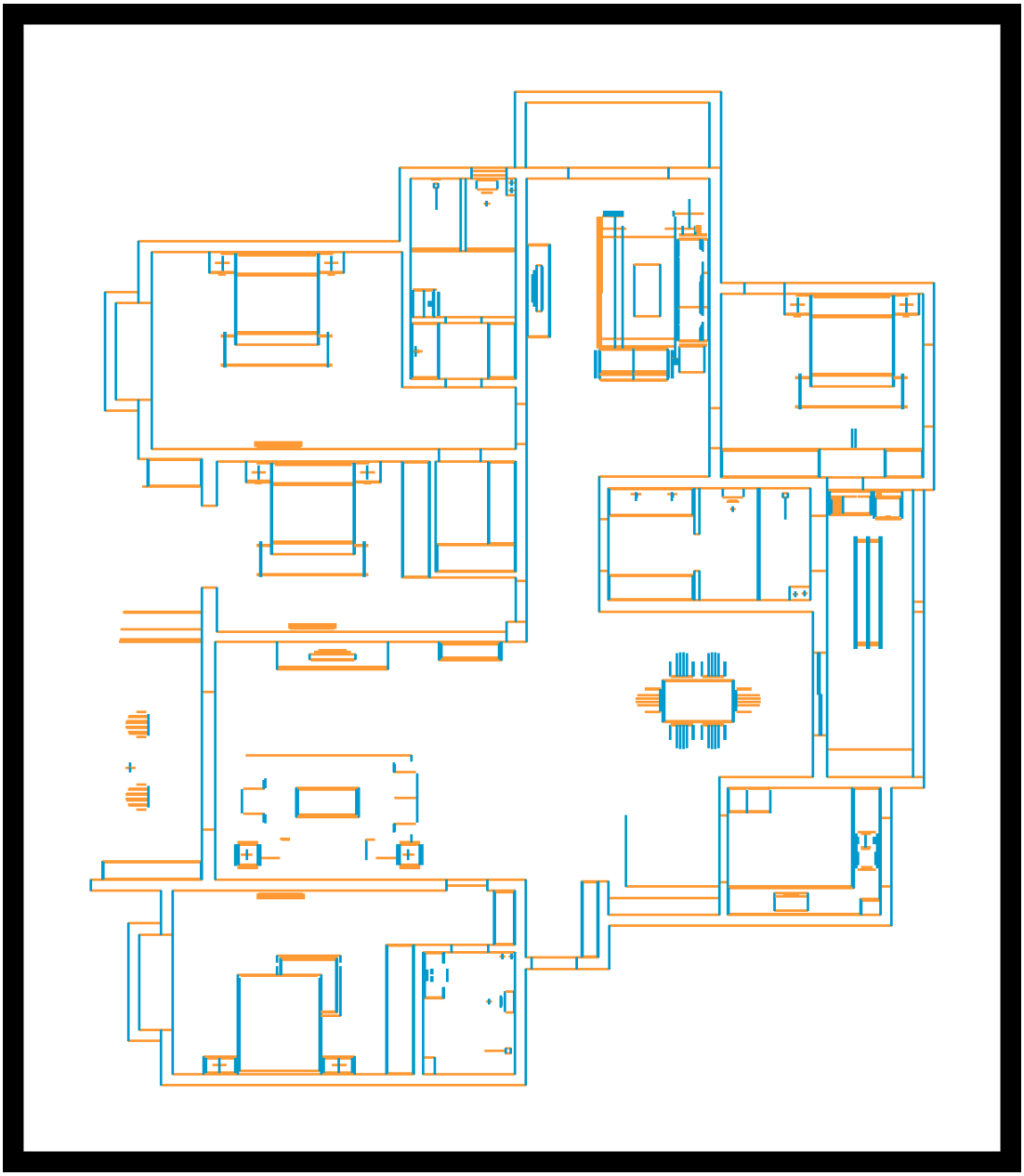


Figure 4-11: *Production from applying a connectivity filter, which removes irrelevant lines*

line-merging filter in the vertical and horizontal directions, respectively. The gap filling filter connects lines in H_{s6} and V_{s6}, if they are sufficiently close.

When this filter is applied to door gaps, the threshold is set to 90 mm, but this is set at 50mm when applying it to line merging. Considering the fact that walls generally are 120mm-240mm in width, setting 50mm as a benchmark will not disturb the results of wall selection.

In Fig 4-12 , H_{s7} and V_{s7} are represented by red and blue lines, respectively.

$$(H_{s7}, V_{s7}) = f_{merge(50mm)}(f_{fill(90mm)}(H_{s6}, V_{s6})) \tag{4.8}$$

8. Detecting Candidate Pairs of Parallel Lines After the second gap-filling and line-merging filters are applied, candidate pairs of parallel lines that contribute to walls are identified and here, two constraints are introduced. One is that the distance between the lines in H_{s7} and V_{s7} should be between 100 mm and 400 mm.

This constraint ensures that walls with width within this range are detected. The other constraint is that the overlapping length between each pair of lines should be greater than 400 mm, because such lines are more likely to form walls. In Equation 4.9 ,H_{s8} and V_{s8} denote the productions of this step in the vertical and horizontal direction respectively. Fig 4-13 shows them as red and blue lines, respectively.

$$(H_{s8}, V_{s8}) = f_{pair}(H_{s7}, V_{s7}) \tag{4.9}$$

9. Generate Walls In this step, walls are generated from the candidate pairs of parallel lines in H_{s8} and V_{s8}. Here the, lines between 100 mm and 400 mm from the target line are found and such lines are considered wall candidates. Then, as shown in Figure 4-14, walls are generated from the overlapping area; however, this method can generate incorrect walls. Hence, such errors have to be fixed in the image-parsing stage (Section 4.3.2).

Image-Parsing Wall Restoration System

With the above filters, whilst the aim is to extract as many correct walls as possible, the input data are difficult to standardise, which inevitably leads to problems. For example, the proposed system may generate incorrect walls or fail

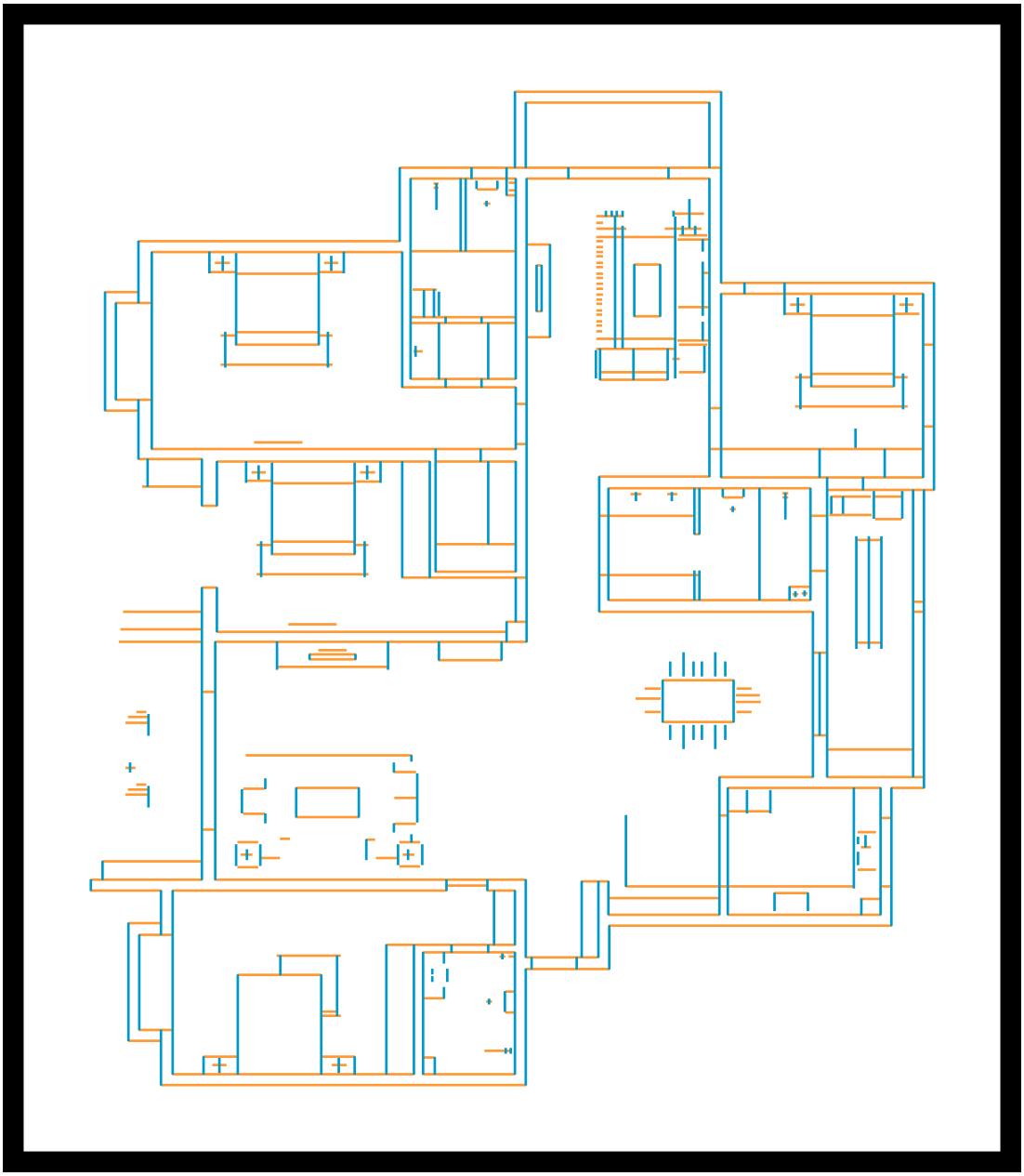


Figure 4-12: *Production of the second fill gap and merge line processing. This is the process of filling gaps between doors and long lines. The red and blue lines represent Hs7 and Vs7 in Equation 4.9, respectively*

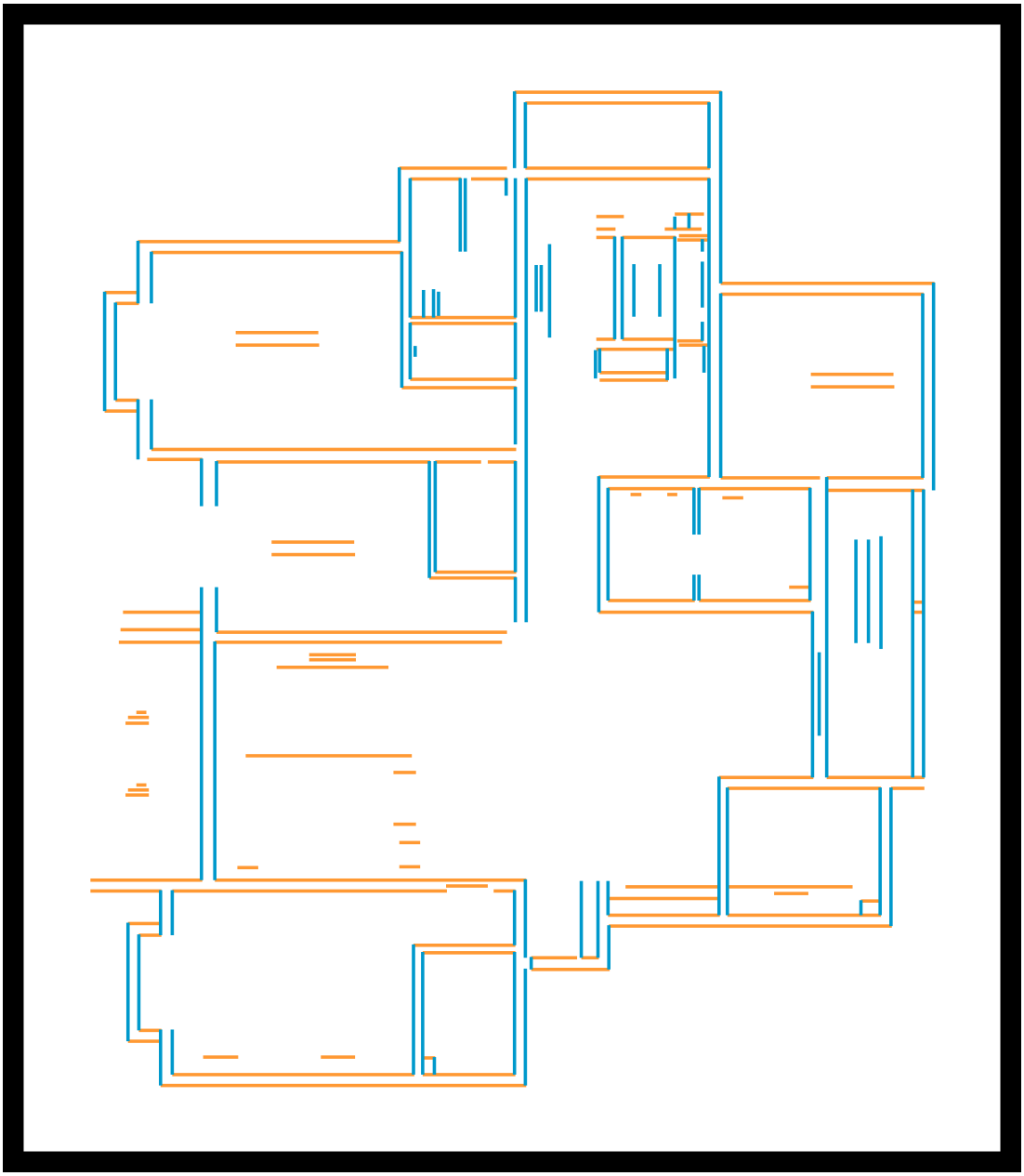


Figure 4-13: *Identify pairs of parallel lines as candidates. Parallel lines in the vertical and horizontal directions are marked as red and blue, respectively*

Figure 4-14: *Walls are generated from the candidate pairs of parallel lines; the overlapping areas that marked in yellow are walls.*

to detect correct ones due to excessive filtering. To address this, a wall restoration method based on image parsing of vectorised CAD floor plans is employed. An image-processing technique is not employed to recognise floor plans in CAD format [103] [135], if the vectorised parsing method is applied. However, the proposed system integrates the filtered results with an image-parsing mechanism. With analysis of the rasterised image, this system evaluates the wall candidates generated from the first step. More specifically, first, the CAD floor plan and the filtered results are rasterised. Then, the components in the floor plan image are extracted by applying an image component segmentation method. The wall restoration method is discussed in Section 4.3.2.

Rasterisation The proposed method converts CAD format floor plans into images. It should be noted that floor plans must be rasterised before the image-parsing method is applied to the vectorised CAD format floor plans. Thus, a raw CAD floor plan and the detected walls are rasterised as images I_{raw} and I_{walls}, respectively, at equal image resolution (i.e. 4096x4096 pixels). Figure 4-15 demonstrates the rasterised results of the raw input data, while Figure 4-16 shows those for walls extracted in the filter steps.

Component Segmentation In floor plans, wall regions can always be clearly distinguished from other objects. Therefore, image segmentation is employed to classify the components in I_{raw}. In addition, I_{raw} is a binary image and hence, the image segmentation task can be converted into a labelling task for connected components in I_{raw}. A connected component in a binary image is a set of pixels that form a connected group. For example, the binary image (left part of Figure 4-17) [122] has three connected components. The connected component labelling process identifies these in the binary image and assigns a unique label to each [122] (right part of Figure 4-17).b. The proposed method employs an eight-connectivity-based two-pass connected component labelling method similar to [73]. The pseudocode of this algorithm is shown in Figure 4-18. The algorithm makes two passes over the image, with the first assigning temporary labels and records equivalences, whilst the second replaces each temporary label with the smallest label of its equivalent class. The following is performed in the first pass:

1. Each element of the data is iterated by column then by row (raster scanning);

2. If the element is not the background;

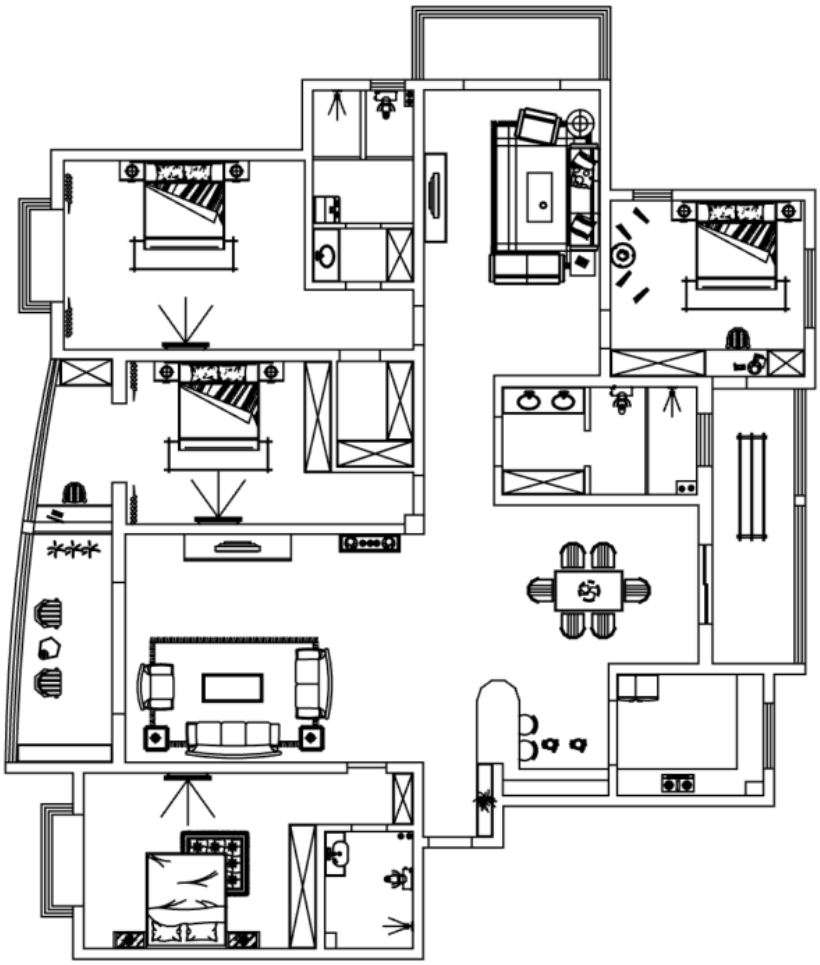


Figure 4-15: *Floor plan must be rasterised before applying the image-parsing method. This figure shows a rasterised result of raw input data*

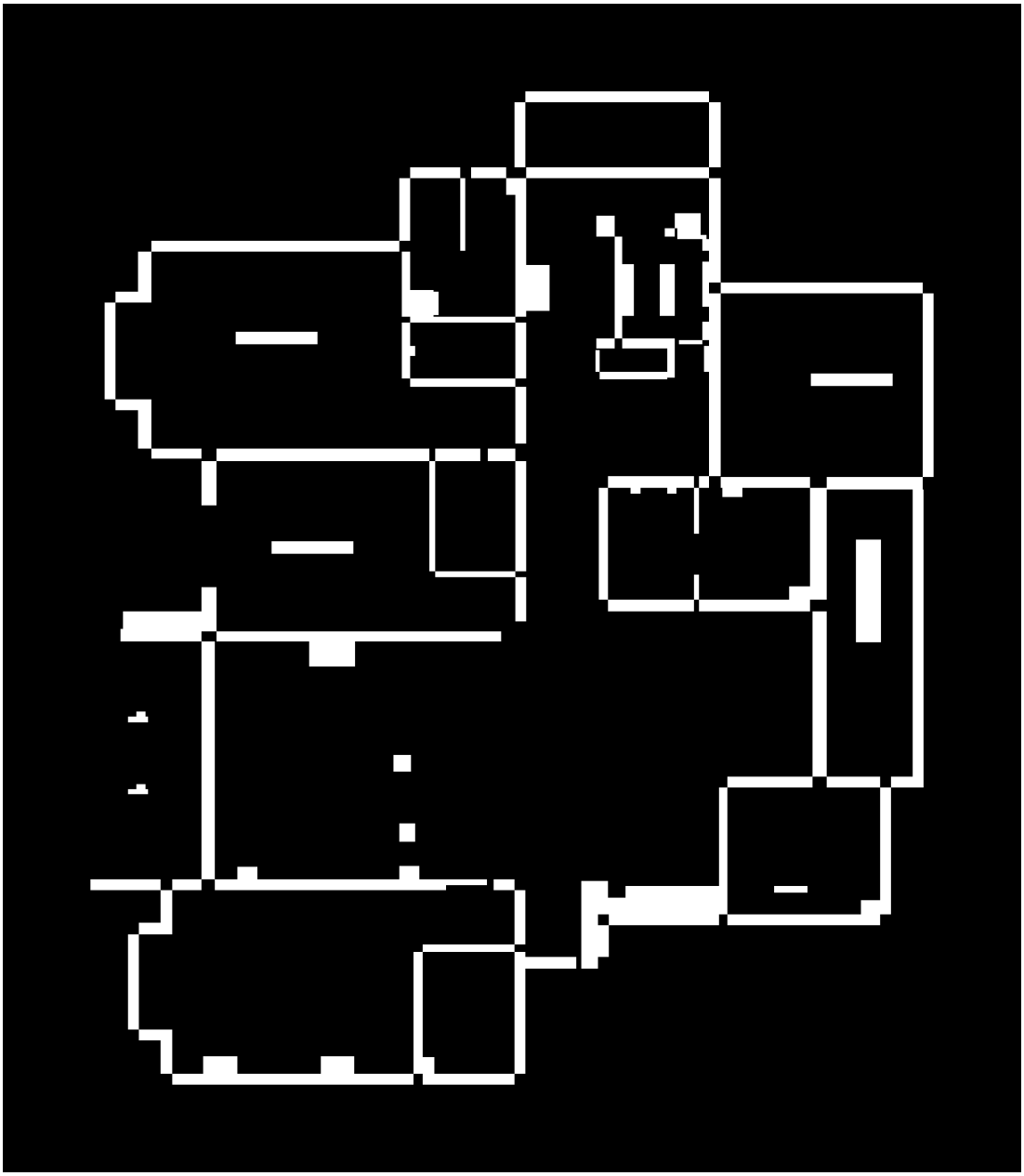


Figure 4-16: *Floor plan must be rasterised before applying the image-parsing method. This figure shows a rasterised result for walls extracted in the filter steps*

Figure 4-17: *Left: An example of connected components in a binary image with three connected components. Right: An example of labelling connected components in a binary image. Connected components in the binary image are identified before being a new unique label*

a. Get the neighbouring elements of the current element;

b. If there are no neighbours, uniquely label the current element and continue;

c. Otherwise, find the neighbour with the smallest label and assign it to the current element;

d. Store the equivalent between neighbouring labels.

The following is performed in the second pass:

1. Iterate through each element of the data by column and then by row;

2. If the element is not the background, relabel it with the lowest equivalent label.

Figure 4-19 shows component labelling result $I_{rawComponent}$.

Wall Restoring The wall regions in a floor plan are distinct from other objects and hence wall information detected in the filter stage can be matched to labelled components in the original image I_{raw}. By tracking each component in the original image and comparing a single component to a wall mask, the component can be defined as a wall, if more than 40% of the wall region overlaps with the wall mask. Figure 4-15 shows the result after a raw process of wall restoration.

However, the product of rough wall restored by wall candidate mask still has some outliers compared to the ground truth and hence, in order to optimise the wall mask, two extra constraints are introduced.

The first filter considers the factor that walls have strong connectivity with each other. Hence, in this filter, unattached regions, which are not connected to

```
algorithm TwoPass(data)
   linked = []
   labels = structure with dimensions of data, initialized with the value of Background

   First pass

   for row in data:
       for column in row:
           if data[row][column] is not Background

               neighbors = connected elements with the current element's value

               if neighbors is empty
                   linked[NextLabel] = set containing NextLabel
                   labels[row][column] = NextLabel
                   NextLabel += 1

               else

                   Find the smallest label

                   L = neighbors labels
                   labels[row][column] = min(L)
                   for label in L
                       linked[label] = union(linked[label], L)

   Second pass

   for row in data
       for column in row
           if data[row][column] is not Background
               labels[row][column] = find(labels[row][column])

   return labels
```

Figure 4-18: *Pseudocode of the connected component labelling algorithm. Temporary equivalent labels are assigned in the first passes and the smallest label of its equivalent class will replace them in the second pass*

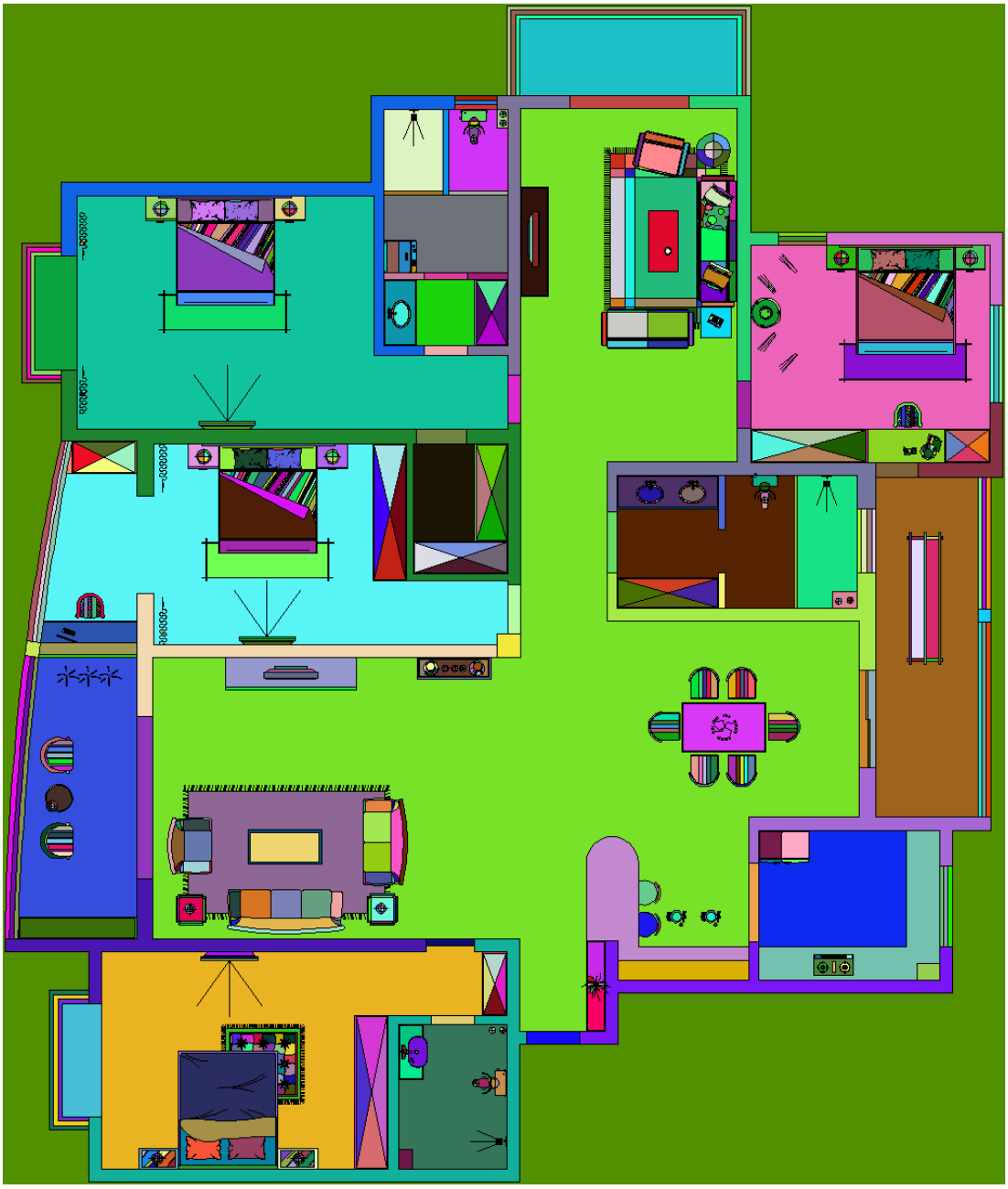


Figure 4-19: *After applying the two-pass algorithm over the image, component labelling result* $I_{rawComponent}$ *generated.*

each other, are detected and removed. More specifically, unattached regions will not be identified as an effective wall, if their area or length and width are less than specific thresholds (in these experiments, the threshold of length and width is 2000mm and 200mm, respectively), and they will be eliminated.

After that, the second constraint takes into account another factor that walls should construct a room. In the real drawing, furniture in the corner that includes a sofa may generate parallel lines as well. Instead of connecting to other lines to generate an independent region, these parallel lines may meet the condition of forming walls. If so, the occupancy rate of the mask is increased to 40% plus, which would be marked as walls. To get around this problem, a limitation needs to be set. If the region takes more than 50% of the rectangle space of that generated by plotting the minimum value and the maximum value on the X and Y axis, there is a tiny possibility of being defined as walls, because these normally do not take up a large area in a CAD drawing. Moreover, in order to avoid interfering the process of detecting small walls, the length and width of the rectangle space are much larger than width of an independent wall. In view of the smallest rooms in the real world, the thresholds are set as 2000mm

Figure 4-20 shows the final wall extraction result obtained by the proposed system.

Detect Windows and Doors

Door Detection Typically, doors are attached to walls, i.e. they do not exist independently and, hence they are identified by detecting arcs that are close to detected walls. Normally, arcs with a 90 degree central angle of radius 300 mm to 25000 mm are considered doors.

Window Detection Window detection is similar to door detection, because windows and doors are both attached to walls. In the proposed method, windows are detected as a part of a wall (Section 4.3.2). Thus, it is possible to find windows by searching multiple parallel lines that are less than 20 mm apart. Also, if the distance between the central line of a group of multiple parallel lines and that of its corresponding wall is less than one-quarter of the wall thickness, this group of multiple parallel lines is considered a window.

Figure 4-20: *Wall restoration is the process of tracking each component in the original image and comparing a single one with a wall mask, and this figure shows the final wall extraction result obtained by the proposed system*

4.4 Evaluation

It is hard to quantitatively evaluate the CAD extraction system because there is barely proper ground truth publicly available. To give an intuitive sense on precision of our system, I implement the proposed architectural drawing recognition system using Java; I then deploy the proposed system onto Kujiale.com which is a Chinese leading Internet company in interior design. In this case, The proposed system, which is freely available online, incorporates a mature human-computer interaction and representation system. Users can obtain recognition results by uploading CAD floor plans (DWT or DWG format). Figure 4-21 shows samples from real customers who actually perform analyzing CAD floor plans for three different real-life projects. Please note that such components extraction is achieved without user intervention.

The proposed system, which is freely available online, incorporates a mature human-computer interaction and representation system. Users can obtain recognition results by uploading CAD floor plans (DWT or DWG format) via the website (www.kujiale.com). On average, the proposed system requires only 2.3s to extract information from a CAD floor plan. Compared to Lu's [103] system, which requires nearly one hour to parse a document with 72,000 graphic primitives, the proposed system requires approximately 5 seconds to parse a complicated floor plan with a massive number of objects, e.g. more than 35,000 lines. The improved efficiency of the proposed system is primarily due to the high-performance algorithm.

As mentioned previously, the proposed system can detect a large proportion of walls in most cases. Because it is intended for practical application by different types of designers, it is difficult to obtain a ground truth for all floor plans. Due to the lack of a CAD floor plan database, the proposed system was evaluated in a user study. Users were asked to score the recognition results of the system (1 means not satisfied and 10 means very satisfied). As shown in Figure 4-22, based on the research, an average score of 7.71 was obtained from 2,515 user study samples, which indicates outstanding performance of the system.

Meanwhile, according to the statistics on CAD recognition system, the number of recognition requests fluctuated at 80,000 each week between March and July in 2017. The figure bottomed at 70,000 on week of 4th April and reached a peak at approximately 98,000 on week of 13th June.

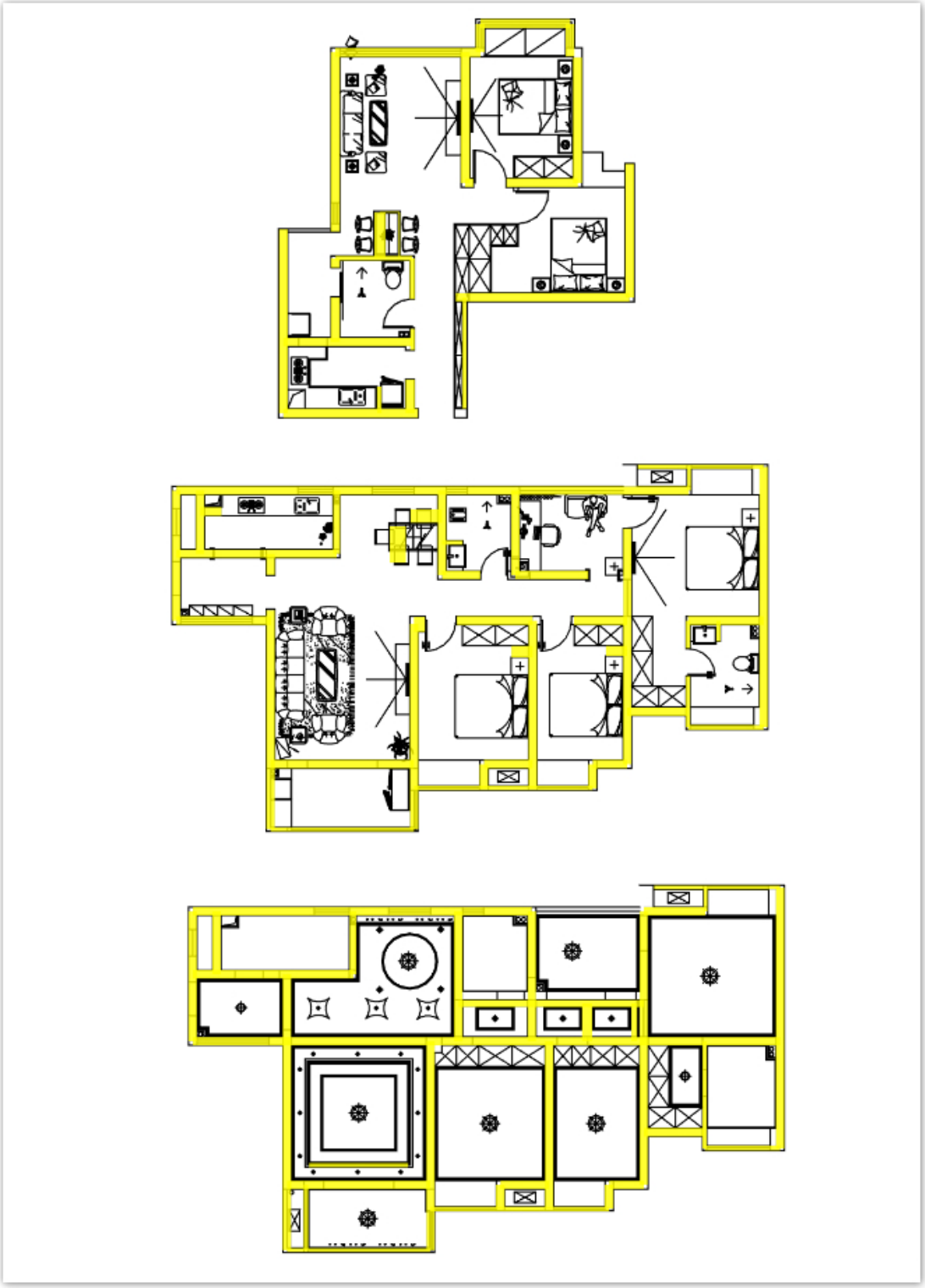


Figure 4-21: *The result of analysing CAD floor plans for three different real-life projects. The evaluation result proves that the system is able to complete the recognition process without user intervention*

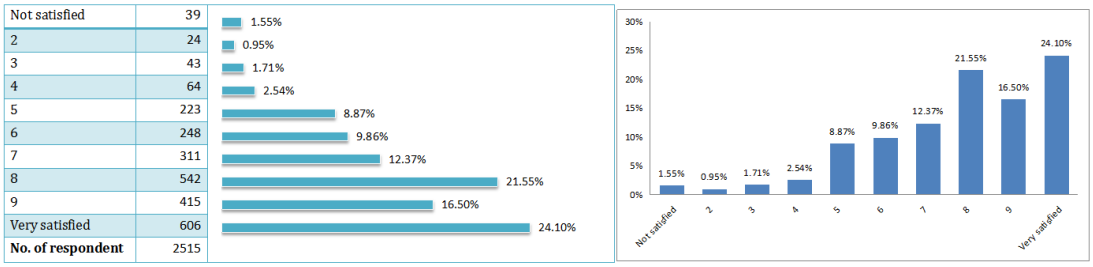


Figure 4-22: *Based on the research, an average score of 7.71 from 2,515 user study samples was obtained, which indicates outstanding performance of the system*

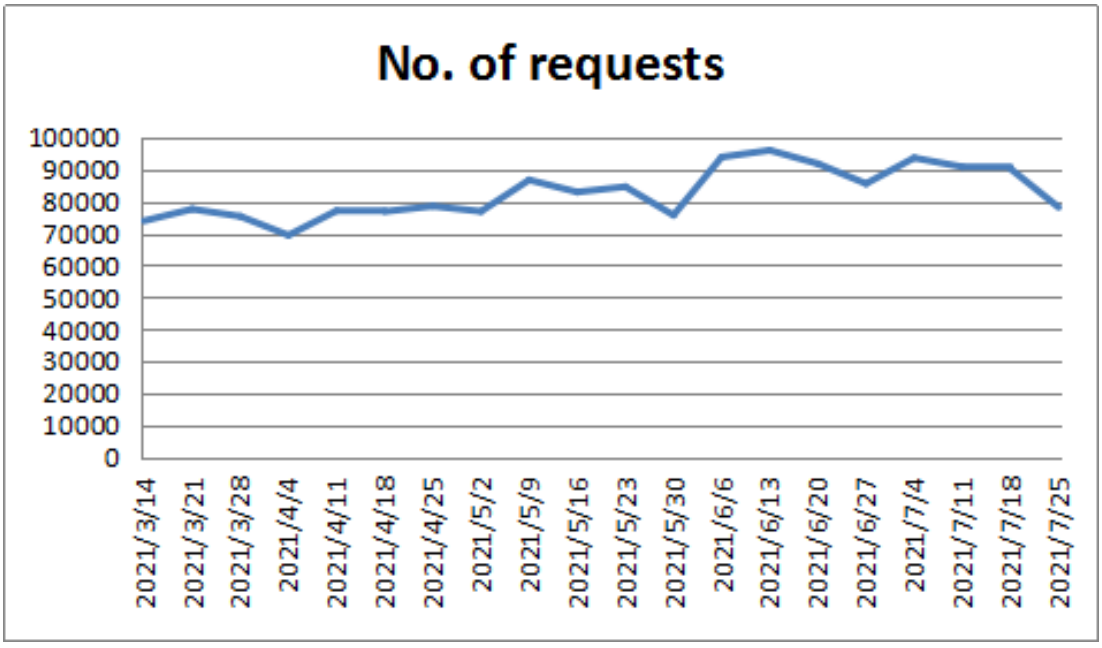


Figure 4-23: *Statistic of the CAD recognition System. There are over ten thousand requests per day*

4.5 Conclusion and Further Work

A complete system for the automatic detection and labelling of architectural elements from CAD format floor plan drawings has been proposed. It consists of structural and semantic analysis processes for identifying relevant information and filtering irrelevant information simultaneously. By applying these processes, useful architectural components, such as walls, windows and doors, can be extracted.

Whilst the theory behind such systems has frequently been discussed over the last two decades, no mature system has been developed for general use. Before the release of the proposed system, manual processes have dominated the market (AutoCAD Revit [123] and Chief Architect [37]). However, the results of a user evaluation (ten thousand uses per day) demonstrate that the proposed system is efficient and effective.

In China, the large population and increasing urbanisation have increased the need for new housing, which has stimulated rapid real estate development. In consideration of the proposed system's contributions, it is believed that it can have significant economic influence by benefiting the architectural industry.

The discussions and evaluations presented in this chapter have demonstrated that the proposed system outperforms an existing method by reducing the processing time to only 2-3s. In the user study, the proposed system achieved an impressive satisfaction rate (90%).

Based on previous discussion and evaluation, the system proposed in this chapter outperform previous method by reducing processing time to just 2-3s. Meanwhile, the system achieves an impressive satisfaction rate that nine out of ten of the targeted users are satisfied with the result.

In future, the aim is to improve the proposed system in several ways. For example, currently, it is weak when detecting arc walls. Thus, further research into detecting arcs more efficiently is planned. Another problem is that, if the user defines the unit in a floor plan incorrectly, the proposed system will fail to extract anything from it and hence, this issue must be addressed in future. Furthermore, constructing 3D models from 2D floor plans is required by both designers and scientists, hence the intention is also to address this issue in the near future.

CHAPTER 5

CONCLUSIONS

5.1 Conclusion and Contribution

For this thesis, the feasibility of applying state-of-the-art image-processing techniques to improve 3D facial mesh alignment and component extraction in CAD floor plans has been studied as well as their potential advantages being evaluated. First, related research and technologies were reviewed. Then, an improved application for registering 3D facial meshes was presented and evaluated. In addition, a fully automatic CAD floor plan regularisation and component extraction system was proposed, which fuses image-processing techniques with a traditional solution.

The contributions of this thesis are summarised as follows.

a. A methodology in which 2D image-processing methods are applied to provide solutions for 3D models was proposed;

b. An algorithm for finding dense correspondences between 3D facial meshes was proposed. The advantages and effectiveness of the proposed algorithm were evaluated experimentally and compared with existing state-of-the-art techniques;

c. A facial expression transfer method was introduced. The purposed method was able to apply facial retargeting from 2D images to 3D meshes in real-time.

d. A fully automatic CAD floor plan regularisation and component extraction system was proposed. Similar to the proposed 3D mesh alignment solution, it fuses image-processing techniques with traditional solutions to improve CAD floor plan regularisation and component extraction performance. A user study was also performed, and the proposed system was compared to an existing one.

5.2 Future Work

The methodology proposed in this thesis has broad versatility. Due to time limitations, for this study the proposed methodology was applied to two specific research areas. However, it is believed that the proposed methodology can be applied to other fields, such as 3D mesh simplification. Previously, similar methodologies have been applied in this way and thus, it is likely that the proposed methodology can be applied to the conversion of 3D structures into 2D UV planes. A 3D mesh simplification algorithm that implements the proposed image-processing techniques is expected to optimise or improve the 2D aspect.

APPENDIX A

LINEAR FITTING

A.1 Introduction

In this section, implementation of an efficient linear method [6] for statistically recovering the full 3D shape of faces from single 2D images is presented. Firstly, the main mechanism of the linear method is explained in theory and the implementation of Aldrian and Smith's linear method for 3D shape reconstruction is demonstrated. Then, the experiments and testing results of the implementation will be presented. Finally, the results and some future works are proposed.

A.2 Linear Shape Fitting Algorithm

People have used the 3D Morphable Model (3DMM) [24] for over ten years. However, lots shape and texture fitting methods of 3DMM are non-linear [25, 125, 116]. Aldrian and Smith's linear solution, which does not need to update shape and texture parameters in iteration is presently the state-of-the-art method for 2D to 3D fitting.

The details of this method are as following (Fig A-1):

a. **Feature Points:** Before starting the algorithm, a sparse set of $N << p$ feature points on the image A-2 is landmarked, where p is the number of 3D scan vertexes. In the current case, all visible Farkas Feature points [45] are selected and six edges-define points.Regarding those feature points in 2D, there are notated them as $x_i \in \mathbb{R}^3$ and for the corresponding 3D locations of 2D feature points, these are marked as homogeneous coordinates $X_i \in \mathbb{R}^4$.

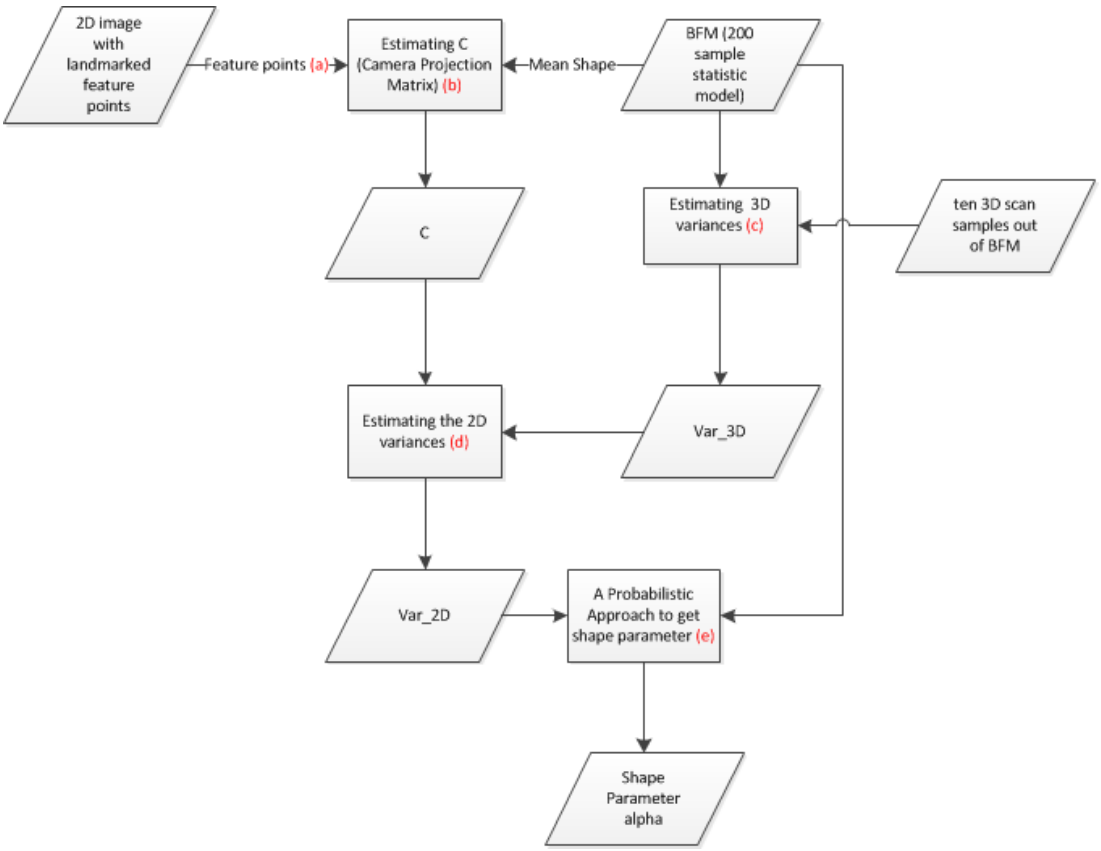


Figure A-1: *Five main steps for the linear shape fitting algorithm*

b. **Estimating Camera Projection Matrix:** In the first step, the corresponding 2D-3D feature points are obtained. Then, the feature points and the mean 3D-shape of the Basel Face Model (BFM) [116] are used to estimate an approximate camera projection matrix C [6]. With the aim of calculating the approximate value of the camera projection matrix C, first, the 2D and 3D feature points are normalised: $\widetilde{x_i} = Tx_i$ and $\widetilde{X_i} = UX_i$, where $T \in \mathbb{R}^{3\times4}$ and $U \in \mathbb{R}^{4\times4}$ translate the centroid of the image/model to the origin and scale them such that the RMS distances from the origin is $\sqrt{3}$ for the 2D points and $\sqrt{3}$ for the 3D ones.

With the assumption that it is an affine camera, the normalised projection matrix, $\widetilde{C} \in \mathbb{R}^{3\times4}$, is computed using the Gold Standard Algorithm čitehartley2000multiple. Given $N \geq 4$ corresponding points between 2D image and 3D model $x_i \leftrightarrow X_i$, the maximum likelihood estimate of $\widetilde{C}$ is determined, which minimises $\sum_i \|\widetilde{x_i} - \widetilde{C}\widetilde{X_i}\|^2$, subject to the affine constraint $\widetilde{C_3} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}$. Each correspondence pair has to follow $2N \times 8$ system of equations:

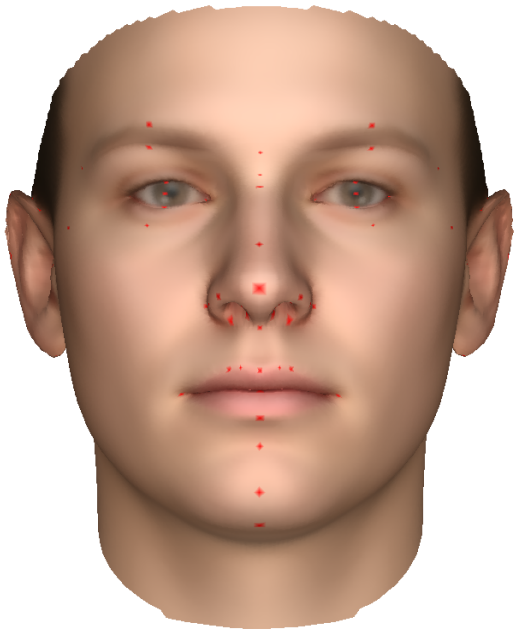


Figure A-2: *Selected feature points in 2D*

$$\begin{bmatrix} \widetilde{X_1}^T & 0^T \\ 0^T & \widetilde{X_1}^T \\ \cdots & \cdots \\ \widetilde{X_N}^T & 0^T \\ 0^T & \widetilde{X_N}^T \end{bmatrix} \times \begin{bmatrix} \widetilde{C_1}^T \\ \widetilde{C_2}^T \end{bmatrix} = \begin{bmatrix} \widetilde{x_{1,1}} \\ \widetilde{x_{1,2}} \\ \cdots \\ \widetilde{x_{N,1}} \\ \widetilde{x_{N,2}} \end{bmatrix} \tag{A.1}$$

Eq A.1 can be simply solved by using least squares, and the camera matrix can be obtained by a de-normalised function $C = T^{-1}\widetilde{C}U$.

c. **Estimating 3D Variances:** According to [24], 3D shape parameters can be recovered by using a probabilistic approach. However, Aldrian and Smith's method need to calculate 2D variances $\sigma^2_{2D,i}$ as well. As long as the 2D variances result can be obtained based on the estimation of 3D variances, ten 3D out-of-sample spaces of BFM will be used in this estimation.Given an out-of-sample face mesh v_i (i.e. a face that was not used to train the sta-

tistical model), it is possible to project onto the BFM to obtain the closest (in a least squares sense) possible approximation of $v_i' = SS^T(v_i - \bar{v}) + \bar{v}$. With a set of (in this case, $k = 10$) vectors of squared errors $e_i = (v_i - v_i')^2$, the variance associated with the coordinates of feature points in unit of mm^2: $\sigma^2_{3D,j} = \frac{1}{k}\sum_{i=1}^{k} \hat{e}_{i,j}$ can be computed, where $\hat{e}_i$ are the sub-elements of e_i, which correspond to the N sparse feature points.

d. **Estimating 2D Variances:** In order to predict the variation in the image plane, the result of 3D variation $\sigma^2_{3D,j}$ to 2D has to be projected in pixels. This can be computed by the following Eq A.2:

$$\begin{bmatrix} \sigma^2_{2D,3i-2} \\ \sigma^2_{2D,3i-1} \\ \sigma^2_{2D,3i} \end{bmatrix} = C_T \times \begin{bmatrix} \sigma^2_{3D,3i-2} \\ \sigma^2_{3D,3i-1} \\ \sigma^2_{3D,3i} \\ 1 \end{bmatrix} + \begin{bmatrix} \eta^2 \\ \eta^2 \\ 0 \end{bmatrix} \tag{A.2}$$

where $C_T \in \mathbb{R}^{3\times4}$ is the camera projection matrix without a translational component and η^2 is an ad hoc 2D pixel error; in this case $\eta = 4$ is used.

e. **A Probabilistic Approach to get shape parameter:** According to [6], the probability of observing the data y for a given c_s is:

$$P(y|c_s) = \prod_{i=1}^{3N} v_N \cdot e^{-\frac{1}{2\sigma^2_{2D,i}}[y_{model2D,i} - y_i]^2} \tag{A.3}$$

where, $y_{model2D,i}$ are the homogeneous coordinates of the 3D feature points projected into 2D and V_N are the selected feature points in 3D. By transforming Eq A.3 to the Error Function,the shape reconstruction error can be defined as below:

$$E = -2 \cdot \log P(C_s|y) = \sum_{i=1}^{3N} \frac{[y_{model2D,i} - y_i]^2}{\sigma^2_{2D,i}} + \|c_s\|^2 + const \tag{A.4}$$

where, $\sigma^2_{2D,i}$ are the 2D variances for each feature points and c_s is the shape parameter. As the aim is to minimise the error, the left side of Eq A.4 is assumed as being zero. Then, the target c_s can be computed by solving the function.

A.3 Experiments and Results

The Basel Face Model (BFM) is used as the dataset and model. Moreover, in the experiments, the visible subset of Farkas Feature points and six self-defined edges points are adopted as landmarks for the 2D images and 3D models. In order to obtain sufficient results, four comprehensive tests have been processed.

A.3.1 Test 1: Ten in sample test and its result

In this test, the front 2D-view of 10 samples are adopted, which are used to estimate 3D variances as the 2D inputs and for each input image, up to 65 feature points are chosen. Then, the 85 most significant Eigen vectors are deployed to reconstruct the faces. Compared with the mean shape reconstruction error (2.8493e+005) through the BFM model (ps. this reconstruction is a 3D to 3D reconstruction), that of this 2D to 3D linear method is 1.3060e+006. Moreover, if the ad hoc 2D pixel error is carefully selected, it can go below 9.0e+006.

Figure A-3: *Ten in the sample test*

A.3.2 Test 2: Five out-of-sample test and its results

The BFM is used to generate 5 random faces and their front views are deployed as 2D input in the second test. Similar to Test 1, for each input image, up to 65 feature points are chosen. Then, the 85 most significant vector modes are used to reconstruct the faces. Compared with the mean shape reconstruction error 8.2747 through the BFM model (ps. this reconstruction is a 3D to 3D reconstruction and the error is low because it is generated from the BFM), that of this 2D to 3D linear method is 5.0590e+005. Moreover, if the ad hoc 2D pixel error is carefully, it can go below 5.0e+005.

A.3.3 Test 3: Pose test and its results

Five faces are used in test one and the face is in 6 poses (0,15,30,45,60, and 75 degrees, respectively). For the lower degree posed images, about 60 visible feature points are chosen and for those of a higher degree at least 50 visible feature points are selected. Then, the 70 most significant Eigen vectors are used to reconstruct the faces. Compared with the mean shape reconstruction error 2.8493e+005 through the BFM model, which of this 2D to 3D linear method is 1.4070e+006.

A.4 Conclusions and Future Works

Referring to the evaluations, it emerges that the accuracy of this linear approach is comparable to a state-of-the-art analysis-by-synthesis algorithm. Also, it is impressively faster (less than two seconds using un-optimised Matlab code versus several minutes [24]). However, manual adjustment for ad hoc parameters is still needed for this technique. Hence, in the future, it is possible to find a solution for the automatic selection of the ad hoc 2D pixel error. Moreover, in this study, the construction result of this algorithm is highly dependent on the selection of 2D images, which means efforts on attaining greater stability will have to be exerted in the future.

BIBLIOGRAPHY

[1] 3DMD. http://www.3dmd.com/.

[2] Agrawal, A., Chellappa, R., and Raskar, R. An algebraic approach to surface reconstruction from gradient fields. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 1, IEEE, pp. 174–181.

[3] Agrawal, A., Raskar, R., and Chellappa, R. What is the range of surface reconstructions from a gradient field? In *Computer Vision–ECCV 2006*. Springer, 2006, pp. 578–591.

[4] Ah-Soon, C., and Tombre, K. Architectural symbol recognition using a network of constraints. *Pattern Recognition Letters 22*, 2 (2001), 231–248.

[5] Ahmed, S., Liwicki, M., Weber, M., and Dengel, A. Improved automatic analysis of architectural floor plans. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on* (2011), IEEE, pp. 864–869.

[6] Aldrian, O., and Smith, W. A. A linear approach of 3d face shape and texture recovery using a 3d morphable model. In *Proceedings of the British Machine Vision Conference, pages* (2010), pp. 75–1.

[7] Allen, B., and Curless, B. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (TOG) 22*, 3 (2003), 587–594.

[8] Amberg, B., Romdhani, S., and Vetter, T. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8.

[9] Anguelov, D., Srinivasan, P., Pang, H. C., Koller, D., Thrun, S., and Davis, J. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *International Conference on Neural Information Processing Systems* (2004), pp. 33–40.

[10] Aoki, Y., Shio, A., Arai, H., and Odaka, K. A prototype system for interpreting hand-sketched floor plans. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on* (1996), vol. 3, IEEE, pp. 747–751.

[11] Baltrusaitis, T., Mahmoud, M., and Robinson, P. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition* (2015), pp. 1–6.

[12] Baltrusaitis, T., Robinson, P., and Morency, L. P. Constrained local neural fields for robust facial landmark detection in the wild. In *IEEE International Conference on Computer Vision Workshops* (2013), pp. 354–361.

[13] Baltrusaitis, T., Robinson, P., and Morency, L. P. Openface: An open source facial behavior analysis toolkit. In *IEEE Winter Conference on Applications of Computer Vision* (2016), pp. 1–10.

[14] Bartlett, M. S., Littlewort, G., Frank, M. G., Lainscsek, C., Fasel, I. R., and Movellan, J. R. Automatic recognition of facial actions in spontaneous expressions. *Journal of multimedia 1*, 6 (2006), 22–35.

[15] Bazzo, J. J., and Lamar, M. V. Recognizing facial actions using gabor wavelets with neutral face average difference. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on* (2004), IEEE, pp. 505–510.

[16] Beeler, T., Bickel, B., Beardsley, P., Sumner, B., and Gross, M. High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 40.

[17] Beeler, T., Hahn, F., Bradley, D., Bickel, B., Beardsley, P., Gotsman, C., Sumner, R. W., and Gross, M. High-quality passive facial performance capture using anchor frames. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 75.

[18] Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence 35*, 12 (2013), 2930–2940.

[19] Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 4 (2002), 509–522.

[20] Benedikt, L., Cosker, D., Rosin, P. L., and Marshall, D. Assessing the uniqueness and permanence of facial actions for use in biometric applications. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 40*, 3 (2010), 449–460.

[21] Besl, P., and McKay, N. A method for registration of 3-d shapes. *IEEE Transactions on pattern analysis and machine intelligence 14*, 2 (1992), 239–256.

[22] Beumier, C., and Acheroy, M. 3d facial surface acquisition by structured light. In *International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging* (1999), Citeseer, pp. 103–106.

[23] Beymer, D., Shashua, A., and Poggio, T. Example based image analysis and synthesis.

[24] Blanz, V., and Vetter, T. A morphable model for the synthesis of 3d faces. In *Computer graphics proceedings, annual conference series* (1999), Association for Computing Machinery SIGGRAPH, pp. 187–194.

[25] Blanz, V., and Vetter, T. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 25*, 9 (2003), 1063–1074.

[26] BMG. http://city.csail.mit.edu/bmg/.

[27] Bookstein, F. Thin-plate splines and the atlas problem for biomedical images. In *Information Processing in Medical Imaging* (1991), Springer, pp. 326–342.

[28] Bradley, D., Heidrich, W., Popa, T., and Sheffer, A. High resolution passive facial performance capture. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 41.

[29] Bregler, C., Bregler, C., Bregler, C., and Bregler, C. Realtime facial animation with on-the-fly correctives. *Acm Transactions on Graphics 32*, 4 (2013), 42.

[30] Breuer, P., Kim, K.-I., Kienzle, W., Scholkopf, B., and Blanz, V. Automatic 3d face reconstruction from single images or video. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on* (2008), IEEE, pp. 1–8.

[31] Brostow, G. J., Hernandez, C., Vogiatzis, G., Stenger, B., and Cipolla, R. Video normals from colored lights. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 10 (2011), 2104–2114.

[32] CADFloorPlan. https://en.wikipedia.org/wiki/3dfloorplan.

[33] Cao, C., Weng, Y., Lin, S., and Zhou, K. 3d shape regression for real-time facial animation. *Acm Transactions on Graphics 32*, 4 (2013), 1–10.

[34] Cao, X., Wei, Y., Wen, F., and Sun, J. Face alignment by explicit shape regression. *International Journal of Computer Vision 107*, 2 (2014), 177–190.

[35] Chang, Y., Vieira, M., Turk, M., and Velho, L. Automatic 3d facial expression analysis in videos. In *Analysis and Modelling of Faces and Gestures*. Springer, 2005, pp. 293–307.

[36] Chew, S. W., Lucey, P., Lucey, S., Saragih, J., Cohn, J. F., Matthews, I., and Sridharan, S. In the pursuit of effective affective computing: The relationship between features and registration. *IEEE*

Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42, 4 (2012), 1006–1016.

[37] CHIEFARCHITECT. https://www.chiefarchitect.com/.

[38] COOTES, T. F., EDWARDS, G. J., AND TAYLOR, C. J. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence 23*, 6 (2001), 681–685.

[39] COOTES, T. F., IONITA, M. C., LINDNER, C., AND SAUER, P. Robust and accurate shape model fitting using random forest regression voting. In *European Conference on Computer Vision* (2012), Springer, pp. 278–291.

[40] COSKER, D., KRUMHUBER, E., AND HILTON, A. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 2296–2303.

[41] CRAIG, K. D., HYDE, S. A., AND PATRICK, C. J. Genuine, suppressed and faked facial behavior during exacerbation of chronic low back pain. *Pain 46*, 2 (1991), 161–171.

[42] CRISTINACCE, D., AND COOTES, T. F. Feature detection and tracking with constrained local models. In *British Machine Vision Conference 2006, Edinburgh, Uk, September* (2006), pp. 929–938.

[43] DALE, K., SUNKAVALLI, K., JOHNSON, M. K., VLASIC, D., MATUSIK, W., AND PFISTER, H. Video face replacement. *Acm Transactions on Graphics 30*, 6 (2011), 1–10.

[44] DARWIN, C. *The expression of the emotions in man and animals*, vol. 526. University of Chicago Press, 1965.

[45] DECARLO, D., METAXAS, D., AND STONE, M. An anthropometric face model using variational techniques. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 67–74.

[46] DEL, G. M., AND COLLE, L. Differences between children and adults in the recognition of enjoyment smiles. *Developmental Psychology 43*, 3 (2007), 796–803.

[47] DI4D. http://www.di3d.com/products/.

[48] Dori, D., and Liu, W. Sparse pixel vectorization: An algorithm and its performance evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 21*, 3 (1999), 202–215.

[49] Dosch, P., and Masini, G. Reconstruction of the 3d structure of a building from the 2d drawings of its floors. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on* (1999), IEEE, pp. 487–490.

[50] Dosch, P., Tombre, K., Ah-Soon, C., and Masini, G. A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition 3*, 2 (2000), 102–116.

[51] Duda, R. O., and Hart, P. E. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM 15*, 1 (1972), 11–15.

[52] Efros, A. A., and Freeman, W. T. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 341–346.

[53] Ekman, P., Friesen, W., and Hager, J. Facial action coding system. *A Human Face* (2002).

[54] Ekman, P., and Friesen, W. V. Constants across cultures in the face and emotion. *Journal of personality and social psychology 17*, 2 (1971), 124.

[55] Ekman, P., and Rosenberg, E. L. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, 1997.

[56] Faggian, N., Paplinski, A., and Sherrah, J. 3d morphable model fitting from multiple views. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on* (2008), IEEE, pp. 1–6.

[57] Fasel, B., and Luettin, J. Automatic facial expression analysis: a survey. *Pattern recognition 36*, 1 (2003), 259–275.

[58] Floyd, R. W. Algorithm 97: Shortest path. *Communications of the Acm 5*, 6 (1962), 345.

[59] Frankot, R. T., and Chellappa, R. A method for enforcing integrability in shape from shading algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 10*, 4 (1988), 439–451.

[60] Friberg, A., B. R., and Sundberg, J. *Analysis by synthesis*, vol. 2. SAGE Publications, Inc., 2014.

[61] Gaffney, S. J. *Probabilistic curve-aligned clustering and prediction with regression mixture models*. PhD thesis, Citeseer, 2004.

[62] Garrido, P., Valgaerts, L., Rehmsen, O., Thormaehlen, T., Perez, P., and Theobalt, C. Automatic face reenactment. In *IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 4217–4224.

[63] Geng, Z. J. Rainbow three-dimensional camera: new concept of high-speed three-dimensional vision systems. *Optical Engineering 35*, 2 (1996), 376–383.

[64] Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 6 (2001), 643–660.

[65] Gibson, J. J. The perception of the visual world.

[66] Goldberger, J. Registration of multiple point sets using the em algorithm. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision* (1999), pp. 730–736 vol.2.

[67] Gong, B., Wang, Y., Liu, J., and Tang, X. Automatic facial expression recognition on a single 3d face by exploring shape deformation. In *Proceedings of the 17th ACM international conference on Multimedia* (2009), ACM, pp. 569–572.

[68] Gonzalez, R., and Woods, R. *Digital Image Processing*. Pearson/Prentice Hall, 2008.

[69] Gosselin, P., Kirouac, G., and Dore, F. Y. Components and recognition of facial expression in the communication of emotion by actors. *Journal of Personality and Social Psychology 68*, 1 (1995), 83.

[70] Gunes, H., and Pantic, M. Automatic, dimensional and continuous emotion recognition. *International Journal of Synthetic Emotions (IJSE) 1*, 1 (2010), 68–99.

[71] Hahnel, D., Thrun, S., and Burgard, W. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. *Ijcai* (2003), 915–920.

[72] Hamm, J., Kohler, C. G., Gur, R. C., and Verma, R. Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. *Journal of neuroscience methods 200*, 2 (2011), 237–256.

[73] Haralock, R. M., and Shapiro, L. G. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[74] Harrigan, J. A., Rosenthal, R., and Scherer, K. R. *The new handbook of methods in nonverbal behavior research*. Oxford University Press, 2008.

[75] Hartley, R., and Zisserman, A. *Multiple view geometry in computer vision*, vol. 2. Cambridge Univ Press, 2000.

[76] Heeger, D. J., and Bergen, J. R. Pyramid based texture analysis and synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM, pp. 229–238.

[77] Hilaire, X., and Tombre, K. Robust and accurate vectorization of line drawings. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 6 (2006), 890–904.

[78] Horn, B. K. P. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A 4*, 4 (1987), 629–642.

[79] Huang, P. S., Hu, Q., Jin, F., and Chiang, F.-P. Color-encoded digital fringe projection technique for high-speed three-dimensional surface contouring. *Optical Engineering 38*, 6 (1999), 1065–1071.

[80] Jarvis, R. Range sensing for computer vision. *Three-dimensional object recognition systems 1* (1993), 17–56.

[81] Jeni, L. A., Girard, J. M., Cohn, J. F., and De La Torre, F. Continuous au intensity estimation using localized, sparse facial feature space. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on* (2013), IEEE, pp. 1–7.

[82] Ji, H. H., Lee, D. D., and Saul, L. K. Learning high dimensional correspondences from low dimensional manifolds. 34–41.

[83] Jurie, F. Solution of the simultaneous pose and correspondence problem using gaussian error model. *Computer Vision and Image Understanding 73*, 3 (1999), 357–373.

[84] Kaltwang, S., Rudovic, O., and Pantic, M. Continuous pain intensity estimation from facial expressions. *Advances in visual computing* (2012), 368–377.

[85] Kemelmacher-Shlizerman, I., and Basri, R. 3d face reconstruction from a single image using a single reference face shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 2 (2011), 394–405.

[86] Kenji, M. Recognition of facial expression from optical flow. *IEICE TRANSACTIONS on Information and Systems 74*, 10 (1991), 3474–3483.

[87] Klaudiny, M., and Hilton, A. High-detail 3d capture and non-sequential alignment of facial performance. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on* (2012), IEEE, pp. 17–24.

[88] Kohler, C. G., Martin, E. A., Stolar, N., Barrett, F. S., Verma, R., Brensinger, C., Bilker, W., Gur, R. E., and Gur, R. C. Static posed and evoked facial expressions of emotions in schizophrenia. *Schizophrenia Research 105*, 1 (2008), 49–60.

[89] Kohler, C. G., Turner, T., Stolar, N. M., Bilker, W. B., Brensinger, C. M., Gur, R. E., and Gur, R. C. Differences in facial expressions of four universal emotions. *Psychiatry Res 128*, 3 (2004), 235–244.

[90] Lam, L., Lee, S.-W., and Suen, C. Y. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence 14*, 9 (1992), 869–885.

[91] Leventon, M. E. Statistical models in medical image analysis. *Massachusetts Institute of Technology* (2000).

[92] Lewis, R., and Séquin, C. Generation of 3d building models from 2d architectural plans. *Computer-Aided Design 30*, 10 (1998), 765–779.

[93] Li, H., and Hartley, R. The 3d-3d registration problem revisited. In *IEEE International Conference on Computer Vision* (2007), pp. 1–8.

[94] Li, H., Yu, J., Ye, Y., and Bregler, C. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics 32*, 4 (July 2013).

[95] Li, S. Z., and Jain, A. K. *Encyclopedia of Biometrics: I-Z.*, vol. 1. Springer Science & Business Media, 2009.

[96] Li, W., Cosker, D., Brown, M., and Tang, R. Optical flow estimation using laplacian mesh energy. In *26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2013).

[97] Lien, J., Kanade, T., Cohn, J., and Li, C. Detection, tracking, and classification of action units in facial expression. *Robotics and Autonomous Systems 31*, 3 (2000), 131–146.

[98] Lin, M. H. Tracking articulated objects in realtime range image sequences. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision* (1999), pp. 648–653 vol.1.

[99] Lladós, J., López-Krahe, J., and Martí, E. A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. *Machine Vision and Applications 10*, 3 (1997), 150–158.

[100] Lladós, J., Martí, E., and Villanueva, J. J. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 10 (2001), 1137–1143.

[101] Loria, P. Symbol recognition contest: A synthesis. pp. 45–49.

[102] Lu, C., and Tang, X. Surpassing human-level face verification performance on lfw with gaussianface. In *AAAI* (2015), pp. 3811–3819.

[103] Lu, T., Yang, H., Yang, R., and Cai, S. Automatic analysis and integration of architectural drawings. *International Journal of Document Analysis and Recognition (IJDAR) 9*, 1 (2007), 31–47.

[104] Luo, P., Wang, X., and Tang, X. A deep sum-product architecture for robust facial attributes analysis. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 2864–2871.

[105] M. Suwa, N. S., and Fujimora, K. A preliminary note on pattern recognition of human emotional expression. *the 4th International Joint Conference on Pattern Recognition* (1978), 408–410.

[106] Ma, W.-C., Jones, A., Chiang, J.-Y., Hawkins, T., Frederiksen, S., Peers, P., Vukovic, M., Ouhyoung, M., and Debevec, P. Facial performance synthesis using deformation-driven polynomial displacement maps. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 121.

[107] Macé, S., Locteau, H., Valveny, E., and Tabbone, S. A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems* (2010), ACM, pp. 167–174.

[108] Mahoor, M. H., Cadavid, S., Messinger, D. S., and Cohn, J. F. A framework for automated measurement of the intensity of non-posed facial action units. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on* (2009), IEEE, pp. 74–80.

[109] Makadia, A., Patterson, A., and Daniilidis, K. Fully automatic registration of 3d point clouds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), pp. 1297–1304.

[110] Moeslund, T. B. *Visual analysis of humans: looking at people*. Springer, 2011.

[111] Mpiperis, I., Malassiotis, S., and Strintzis, M. G. Bilinear models for 3-d face and facial expression recognition. *Information Forensics and Security, IEEE Transactions on 3*, 3 (2008), 498–511.

[112] Myronenko, A., and Song, X. Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 12 (2010), 2262–2275.

[113] Or, S.-H., Wong, K.-H., Yu, Y.-k., Chang, M. M., and Kong, H. Highly automatic approach to architectural floorplan image understanding & model generation. *Pattern Recognition* (2005), 25–32.

[114] Pantic, M., Nijholt, A., Pentland, A., and Huanag, T. S. Human-centred intelligent human? computer interaction: how far are we from attaining it? *International Journal of Autonomous and Adaptive Communications Systems 1*, 2 (2008), 168–187.

[115] Patel, A., and Smith, W. 3d morphable face models revisited. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 1327–1334.

[116] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. A 3d face model for pose and illumination invariant face recognition.

[117] Prkachin, K. M. The consistency of facial expressions of pain: A comparison across modalities. *Pain 51*, 3 (1992), 297–306.

[118] Prkachin, K. M., and Mercer, S. R. Pain expression in patients with shoulder pathology: validity, properties and relationship to sickness impact. *Pain 39*, 3 (1989), 257–265.

[119] Pudil, P., and Novovicova, J. *Novel Methods for Feature Subset Selection with Respect to Problem Knowledge*. IEEE Educational Activities Department, 1998.

[120] Ravikumar, S., Davidson, C., Kit, D., Campbell, N., Benedetti, L., and Cosker, D. Reading between the dots: Combining 3d markers and facs classification for high-quality blendshape facial animation. In *Graphics Interface* (2016), pp. 143–151.

[121] Reed, L. I., Sayette, M. A., and Cohn, J. F. Impact of depression on response to comedy: A dynamic facial coding analysis. *Journal of Abnormal Psychology 116*, 4 (2007), 804.

[122] revit. http://uk.mathworks.com/help/images/.

[123] revit. www.autodesk.co.uk.

[124] Rocha, E. M., Prkachin, K. M., Beaumont, S. L., Hardy, C. L., and Zumbo, B. D. Pain reactivity and somatization in kindergarten age children. *J Pediatr Psychol 28*, 1 (2003), 47–57.

[125] Romdhani, S., and Vetter, T. Efficient, robust and accurate fitting of a 3d morphable model. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (2003), IEEE, pp. 59–66.

[126] Roweis, S. T., and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (2000), 2323.

[127] Sandbach, G., Zafeiriou, S., and Pantic, M. Binary pattern analysis for 3d facial action unit detection.

[128] Sandbach, G., Zafeiriou, S., Pantic, M., and Yin, L. Static and dynamic 3d facial expression recognition: A comprehensive survey. *Image and Vision Computing* (2012).

[129] Savran, A., Sankur, B., and Bilge, M. T. Regression-based intensity estimation of facial action units. *Image and Vision Computing 30*, 10 (2012), 774–784.

[130] Schürmann, J. *Pattern classification: a unified view of statistical and neural approaches*. Wiley Online Library, 1996.

[131] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 1, IEEE, pp. 519–528.

[132] Shelton, C. R. Morphable surface models. *International Journal of Computer Vision 38*, 1 (2000), 75–91.

[133] Sibbing, D., Habbecke, M., and Kobbelt, L. Markerless reconstruction of dynamic facial expressions. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 1778–1785.

[134] Simchony, T., Chellappa, R., and Shao, M. Direct analytical methods for solving poisson equations in computer vision problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 12*, 5 (1990), 435–446.

[135] So, C., Baciu, G., and Sun, H. Reconstruction of 3d virtual buildings from 2d architectural floor plans. In *Proceedings of the ACM symposium on Virtual reality software and technology* (1998), ACM, pp. 17–23.

[136] Sumner, R. W. Deformation transfer for triangle meshes. In *ACM SIGGRAPH* (2004), pp. 399–405.

[137] Sun, Y., Chen, Y., Wang, X., and Tang, X. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems* (2014), pp. 1988–1996.

[138] Sun, Y., Wang, X., and Tang, X. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2013), pp. 3476–3483.

[139] Tang, R., Cosker, D., and Li, W. Global alignment for dynamic 3d morphable model construction. *WORKSHOP ON VISION AND LANGUAGE 2012 (VL'12)* (2012).

[140] Tenenbaum, J. B., Silva, V. D., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319.

[141] Thies, J., Zollh, Fer, M., Nie, Ner, M., Valgaerts, L., Stamminger, M., and Theobalt, C. Real-time expression transfer for facial reenactment. *Acm Transactions on Graphics 34*, 6 (2015), 1–14.

[142] Tian, Y., Kanade, T., and Cohn, J. F. Recognizing action units for facial expression analysis. *IEEE Trans Pattern Anal Mach Intell 23*, 2 (2001), 97.

[143] Tian, Y.-l., Kanade, T., and Cohn, J. F. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on* (2002), IEEE, pp. 229–234.

[144] Tsalakanidou, F., Forster, F., Malassiotis, S., and Strintzis, M. G. Real-time acquisition of depth and color images using structured light and its application to 3d face recognition. *Real-Time Imaging 11*, 5 (2005), 358–369.

[145] Valstar, M. F., Almaev, T., Girard, J. M., and Mckeown, G. Fera 2015 - second facial expression recognition and analysis challenge. In *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition* (2016), pp. 1–8.

[146] Valstar, M. F., Almaev, T., Girard, J. M., McKeown, G., Mehu, M., Yin, L., Pantic, M., and Cohn, J. F. Fera 2015-second facial expression recognition and analysis challenge. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on* (2015), vol. 6, IEEE, pp. 1–8.

[147] Valstar, M. F., Sánchez-Lozano, E., Cohn, J. F., Jeni, L. A., Girard, J. M., Zhang, Z., Yin, L., and Pantic, M. Fera 2017-addressing head pose in the third facial expression recognition and analysis challenge. *arXiv preprint arXiv:1702.04174* (2017).

[148] Valveny, E., and Marti, E. A model for image generation and symbol recognition through the deformation of lineal shapes. *Pattern Recognition Letters 24*, 15 (2003), 2857–2867.

[149] Viola, P., and Wells, William M., I. *Alignment by maximization of mutual information*. IEEE Computer Society, 1997.

[150] Wang, C., and Mahadevan, S. Manifold alignment using procrustes analysis. In *International Conference on Machine Learning* (2008), pp. 1120–1127.

[151] Wang, S.-F., and Lai, S.-H. Reconstructing 3d face model with associated expression deformation from a single face image via constructing

a low-dimensional expression deformation manifold. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 10 (2011), 2115–2121.

[152] Weber, M., Liwicki, M., and Dengel, A. A. scatch-a sketch-based retrieval for architectural floor plans. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on* (2010), IEEE, pp. 289–294.

[153] Weise, T., Bouaziz, S., Li, H., and Pauly, M. Realtime performance-based facial animation. In *Acm Siggraph* (2011), pp. 1–10.

[154] Wessel, R., Blümel, I., and Klein, R. The room connectivity graph: Shape retrieval in the architectural domain.

[155] Wilson, C. A., Alexander, O., Tunwattanapong, B., Peers, P., Ghosh, A., Busch, J., Hartholt, A., and Debevec, P. Facial cartography: interactive scan correspondence. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), ACM, pp. 205–214.

[156] Wolf, L., Shashua, A., and Wexler, Y. Join tensors: On 3d-to-3d alignment of dynamic sets. In *International Conference on Pattern Recognition* (2000), p. 1388.

[157] Woodham, R. J. Photometric method for determining surface orientation from multiple images. *Optical engineering 19*, 1 (1980), 191139–191139.

[158] Yan, L., and Wenyin, L. Engineering drawings recognition using a case-based approach. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on* (2003), IEEE, pp. 190–194.

[159] Yang, P., Liu, Q., and Metaxas, D. N. Boosting encoded dynamic features for facial expression recognition. *Pattern Recognition Letters 30*, 2 (2009), 132–139.

[160] Yang, S. Symbol recognition via statistical integration of pixel-level constraint histograms: A new descriptor. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2 (2005), 278–281.

[161] Yin, L., Chen, X., Sun, Y., Worm, T., and Reale, M. A high-resolution 3d dynamic facial expression database. In *Automatic Face &*

Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on (2008), IEEE, pp. 1–6.

[162] Yin, L., Wei, X., Longo, P., and Bhuvanesh, A. Analyzing facial expressions using intensity-variant 3d data for human computer interaction. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (2006), vol. 1, IEEE, pp. 1248–1251.

[163] Yin, X., Wonka, P., and Razdan, A. Generating 3d building models from architectural drawings: A survey. *IEEE Computer Graphics and Applications*, 1 (2009), 20–30.

[164] Zeng, Z., Pantic, M., Roisman, G. I., and Huang, T. S. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence 31*, 1 (2009), 39–58.

[165] Zhang, S., and Yau, S.-T. High-resolution, real-time 3d absolute coordinate measurement based on a phase-shifting method. *Optics Express 14*, 7 (2006), 2644–2649.

[166] Zhang, Z., Luo, P., Chen, C. L., and Tang, X. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision* (2014), pp. 94–108.

[167] Zhu, X., and Ramanan, D. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), IEEE, pp. 2879–2886.

[168] Zhu, Z., Luo, P., Wang, X., and Tang, X. Deep learning identity-preserving face space. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 113–120.

[169] Zhu, Z., Luo, P., Wang, X., and Tang, X. Deep learning multi-view representation for face recognition. *arXiv preprint arXiv:1406.6947* (2014).